fli4l – floppy-isdn4linux
Version 3.6.2

Das fli4l-Team
E-Mail: team@fli4l.de

16. September 2012

Inhaltsverzeichnis

1. Einleitung **10**

2. Installation und Konfiguration **13**
2.1. Entpacken der Archive . . . 13
2.2. Konfiguration . . . 14
2.2.1. Editieren der Konfigurationsdateien . . . 14
2.2.2. Konfiguration über eine spezielle Konfigurationsdatei . . . 15
2.3. Installationsvarianten . . . 15
2.3.1. Router auf nur einer Diskette, CD oder vom Netzwerk . . . 16
2.3.2. Typ A: Router auf Festplatte – nur eine FAT-Partition . . . 16
2.3.3. Typ B: Router auf Festplatte – je eine FAT- und ext3-Partition . . . 16

3. Basiskonfiguration **18**
3.1. Beispiel-Datei . . . 19
3.2. Allgemeine Einstellungen . . . 25
3.3. Einbinden von zusätzlicher Diskette . . . 29
3.4. Konsolen-Einstellungen . . . 30
3.5. Loggen der Bootsequenz und des Ladens von Modulen . . . 30
3.6. Verwendung einer eigenen /etc/inittab . . . 31
3.7. Länderspezifische Tastaturlayouts . . . 32
3.8. Ethernet/Token-Ring-Netzwerkkarten-Treiber . . . 32
3.9. Netzwerke . . . 38
3.10. Zusätzliche Routen (optional) . . . 40
3.11. Konfiguration des Paketfilters . . . 41
3.12. Die alte Konfiguration des Paketfilters . . . 41
3.12.1. Kommunikation über den Router hinweg . . . 42
3.12.2. Kommunikation mit dem Router . . . 47
3.13. Neue Konfiguration des Paketfilters . . . 49
3.13.1. Aktionen des Paketfilters . . . 50
3.13.2. Einschränkungen in den Regeln . . . 51
3.13.3. Der Einsatz von Templates im Paketfilter . . . 53
3.13.4. Die Konfiguration des Paketfilters . . . 57
3.13.5. Beispiele . . . 62
3.13.6. Standardkonfigurationen . . . 66
3.13.7. DMZ – Demilitarisierte Zone . . . 71
3.13.8. Port-Forwarding . . . 73
3.13.9. Masquerading Module . . . 75
3.14. Domain-Konfiguration . . . 76
3.15. imond-Konfiguration . . . 78

3.16. Allgemeine Circuit-Konfiguration 80

4. Pakete **81**

4.1. Werkzeuge im Basispaket 81
4.1.1. OPT_SYSLOGD – Protokollieren von Systemmeldungen 81
4.1.2. OPT_KLOGD – Protokollieren von Kernelmeldungen 83
4.1.3. OPT_LOGIP – Protokollieren von WAN-IP-Adressen 83
4.1.4. OPT_Y2K – Datumskorrektur bei nicht Y2K-festen Rechnern 83
4.1.5. OPT_PNP – Installation von isapnp tools 84
4.2. Advanced Networking 86
4.2.1. Bonding - mehrere Netzwerkkarten zusammenfassen zu einem Link . . . 86
4.2.2. VLAN - 802.1Q Unterstüzung 90
4.2.3. Device MTU - Anpassen der MTU 92
4.2.4. BRIDGE - Ethernet Bridging für fli4l 92
4.2.5. Anmerkungen 96
4.2.6. EBTables - EBTables für Fli4l 96
4.2.7. Beispiel 96
4.3. CHRONY - Network Time Protocol Server/Client 98
4.3.1. Konfiguration des OPT_CHRONY 99
4.3.2. Support 100
4.3.3. Literatur 100
4.4. DHCP_CLIENT - Dynamic Host Configuration Protocol 100
4.4.1. OPT_DHCP_CLIENT 100
4.5. DNS_DHCP - Hostnamen, DNS- und DHCP-Server sowie DHCP-Relay 101
4.5.1. Hostnamen 101
4.5.2. DNS-Server 103
4.5.3. DHCP-Server 106
4.5.4. DHCP-Relay 108
4.5.5. TFTP-Server 109
4.6. DSL - DSL über PPPoE, Fritz!DSL und PPTP 109
4.6.1. Allgemeine Konfigurationsvariablen 110
4.6.2. OPT_PPPOE - DSL über PPPoE 113
4.6.3. OPT_PPPOE_CIRC - Mehrere DSL-Circuits über PPPoE (Experimentell) 115
4.6.4. OPT_FRITZDSL - DSL per Fritz!Card DSL 115
4.6.5. OPT_PPTP - DSL über PPTP in Österreich/die Niederlande (EXPERIMENTAL) 116
4.6.6. OPT_POESTATUS - PPPoE-Status-Monitor auf fli4l-Console 118
4.6.7. OPT_PFC - Packet-Filter-Compiler 118
4.7. DYNDNS - Update für DynamicDNS-Dienste 118
4.8. EASYCRON - Befehle zeitgesteuert ausführen 123
4.8.1. Konfiguration 123
4.8.2. Beispiele 124
4.8.3. Voraussetzungen 124
4.8.4. Installation 125
4.9. HD - Unterstützung von Festplatten, Flash-Karten, USB-Sticks, ... 125
4.9.1. OPT_HDINSTALL - Installation auf Festplatte/CompactFlash 125

4.9.2. OPT_MOUNT - Mounten von Dateisystemen auf Festplatte etc. 128
4.9.3. OPT_EXTMOUNT - Mounten von Dateisystemen auf Festplatte etc. . 128
4.9.4. OPT_HDSLEEP – automatisches Abschalten für IDE-Festplatten einstellen 129
4.9.5. OPT_RECOVER – Notfalloption 129
4.9.6. OPT_HDDRV - Treiber für Festplattencontroller 130
4.10. HTTPD - Status-Webserver 131
4.10.1. OPT_HTTPD - Mini-Webserver als Statusmonitor 131
4.10.2. Nutzerverwaltung 133
4.10.3. OPT_OAC - Online Access Control 134
4.11. IPv6 - Internet Protokoll Version 6 135
4.11.1. Einleitung 135
4.11.2. Adressformat 136
4.11.3. Konfiguration 137
4.11.4. Web-GUI 145
4.12. ISDN - Kommunikation über aktive und passive ISDN-Karten 145
4.12.1. Herstellen einer ISDN-Verbindung 145
4.12.2. ISDN-Karte 146
4.12.3. OPT_ISDN_COMP (EXPERIMENTAL) 150
4.12.4. ISDN-Circuits 150
4.12.5. OPT_TELMOND - telmond-Konfiguration 159
4.13. LCD - Anzeige von Statusinformationen über LC-Display 162
4.13.1. Einleitung 162
4.13.2. Konfiguration 162
4.13.3. isdn_rate 167
4.13.4. Anschlußbelegung eines LCD-Moduls am Parallelport 168
4.13.5. Anschluß eines 4x40 Displays 168
4.13.6. Winamp-Verdrahtung eines LCD-Moduls 169
4.13.7. Tips und Tricks - Zusammengefasst aus Beiträgen von Robert Resch . . 170
4.13.8. Danksagung 171
4.14. LPDSRV - Berkeley lpd Unterstützung 172
4.14.1. LPDSRV - Druckerserver für lpr/lpd-Protokoll 172
4.14.2. Konfiguration 173
4.14.3. Einrichtung eines Linux-LPR-Clients 179
4.14.4. Einrichtung eines Windows NT 4.0/2000-LPR-Clients 181
4.14.5. Einrichtung eines Windows 95/98/Me-LPR-Clients 182
4.14.6. Einrichtung eines Mac-Clients (MacOSX 10.3.2) 184
4.14.7. Fehlersuche und Beseitigung 184
4.15. OpenVPN - VPN-Support 185
4.15.1. OpenVPN - Einführendes Beispiel 186
4.15.2. OpenVPN - Konfiguration 187
4.15.3. OpenVPN - Bridgekonfiguration 190
4.15.4. OpenVPN - Tunnelkonfiguration 191
4.15.5. OpenVPN - Delegation von DNS und Reverse-DNS 193
4.15.6. Experteneinstellungen 193
4.15.7. OpenVPN - WebGUI 202
4.15.8. OpenVPN - Zusammenarbeit unterschiedlicher OpenVPN Versionen . . 205

4.15.9. OpenVPN - Beispiele . . . 206
4.15.10.Weiterführende Links zum Thema OpenVPN . . . 210
4.16. PCMCIA - PC-Card Unterstützung . . . 210
4.16.1. PCMCIA-Treiber . . . 210
4.17. PPP - Anbindung eines Rechners über serielle Schnittstelle . . . 211
4.18. PROXY - Verschiedene Proxy-Server . . . 213
4.18.1. OPT_PRIVOXY - Ein Werbung-filternder HTTP-Proxy . . . 213
4.18.2. OPT_TOR - Ein anonymes Kommunikationssystem für das Internet . . . 216
4.18.3. OPT_SS5 - Ein Socks4/5 Proxy . . . 217
4.18.4. OPT_TRANSPROXY (EXPERIMENTELL) - Transparanter HTTP-Proxy . . . 218
4.19. QoS - Quality of Service . . . 219
4.19.1. Konfiguration . . . 219
4.19.2. Anwendungsbeispiele . . . 227
4.20. SSHD - Secure Shell, Secure Copy . . . 234
4.20.1. Installation des Secure-Shell-Dienstes . . . 234
4.20.2. Installation des Secure-Copy-Dienstes . . . 238
4.20.3. Installation des dbclients . . . 238
4.20.4. Installation des plink Clients . . . 238
4.20.5. Installation des sftp-server . . . 239
4.20.6. Literatur . . . 239
4.21. TOOLS - Zusätzliche Werkzeuge zum Debugging . . . 239
4.21.1. Die Hardware-Erkennung (Experimentell) . . . 239
4.21.2. Die Tools . . . 240
4.22. UMTS - Anbindung mittels UMTS an das Internet . . . 244
4.22.1. Konfiguration . . . 244
4.22.2. Beispielkonfiguration für RRDTOOL . . . 247
4.23. USB - Support für USB-Geräte . . . 247
4.23.1. Probleme mit USB-Geräten . . . 249
4.23.2. Hinweise zur Benutzung . . . 249
4.23.3. Mounten von USB-Geräten . . . 249
4.24. WLAN - Wireless-LAN Unterstützung . . . 250
4.24.1. Hinweise zum Prism54 Treiber . . . 250
4.24.2. WLAN-Konfiguration . . . 250
4.24.3. Beispiele . . . 255
4.24.4. rrdtool Integration . . . 257
4.24.5. Spendenhinweis . . . 257
4.25. SRC - Das fli4l Buildroot . . . 257
4.25.1. Eine Übersicht über die Sourcen . . . 258
4.25.2. Das fbr Image vorbereiten . . . 258
4.25.3. Mit dem fbr arbeiten . . . 259
4.25.4. Die Programme der fli4l-Distribution übersetzen . . . 260
4.25.5. Eigene Programme ins fbr einbinden . . . 260
4.25.6. Der Kernel . . . 261

5. Erzeugen der fli4l Archive/Bootmedien **262**
5.1. Erzeugen der fli4l Archive/Bootmedien unter Linux bzw. anderen Unix-Derivaten 262

5.1.1. Kommandozeilenoptionen . . . 263
5.2. Erzeugen der fli4l Archive/Bootmedien unter Windows . . . 265
5.2.1. Kommandozeilenoptionen . . . 265
5.2.2. Konfigurationsdialog – Einstellung des Konfigurationsverzeichnis . . . 266
5.2.3. Konfigurationsdialog – allgemeine Einstellungen . . . 267
5.2.4. Konfigurationsdialog – Einstellungen für Diskette . . . 268
5.2.5. Konfigurationsdialog – Einstellungen für Remoteupdate . . . 269
5.2.6. Konfigurationsdialog – Einstellungen für HD-pre-install . . . 270
5.3. Steuerungsdatei mkfli4l.conf . . . 271
5.4. Erstellung der Bootdiskette . . . 272

6. Anbindung von PCs im LAN **274**
6.1. IP-Adresse . . . 274
6.2. Rechnername und Domain . . . 274
6.2.1. Windows 2000 . . . 274
6.2.2. NT 4.0 . . . 275
6.2.3. Win95/98 . . . 275
6.3. Gateway . . . 275
6.4. DNS-Server . . . 276

7. Client-/Server-Schnittstelle imond **277**
7.1. imon-Server imond . . . 277
7.1.1. Least-Cost-Routing – Funktionsweise . . . 277
7.1.2. Zur Berechnung der Onlinekosten . . . 282
7.2. Windows-Client imonc.exe . . . 282
7.2.1. Einleitung . . . 282
7.2.2. Startparameter . . . 283
7.2.3. Seite Überblick . . . 285
7.2.4. Config-Dialog . . . 286
7.2.5. Seite Anrufe . . . 292
7.2.6. Seite Verbindungen . . . 292
7.2.7. Seite Fax . . . 293
7.2.8. Seite E-Mail . . . 293
7.2.9. Admin . . . 294
7.2.10. Seiten Fehler, Syslog und Firewall . . . 295
7.2.11. Seite News . . . 295
7.3. Unix/Linux-Client imonc . . . 295

8. Entwickler-Dokumentation **298**
8.1. Inkompatibilitäten zwischen 2.0 und 3.x . . . 298
8.2. Allgemeine Regeln . . . 298
8.3. Compilieren von Programmen . . . 299
8.4. Verwendung eines modifizierten Kernels — Compilieren eigener Module . . . 299
8.5. Modulkonzept . . . 301
8.5.1. mkfli4l . . . 301
8.5.2. Aufbau . . . 301
8.5.3. Die Konfiguration der Pakete . . . 303

8.5.4. Die Liste der zu kopierenden Dateien . . . 303
8.5.5. Die Prüfung der Konfiguration-Variablen . . . 306
8.5.6. Eigene Definitionen zum Prüfen der Konfigurationsvariablen . . . 307
8.5.7. Erweiterte Prüfungen der Konfiguration . . . 313
8.5.8. Unterstützung von Linux 2.4 und 2.6 . . . 321
8.5.9. Dokumentation . . . 322
8.5.10. Dateiformate . . . 324
8.5.11. Entwickler-Dokumentation . . . 324
8.5.12. Client-Programme . . . 324
8.5.13. Quellcode . . . 324
8.5.14. Weitere Dateien . . . 324
8.6. Allgemeine Skript-Erstellung auf FLI4L . . . 325
8.6.1. Aufbau . . . 325
8.6.2. Umgang mit Konfigurationsvariablen . . . 325
8.6.3. Fehlersuche . . . 327
8.6.4. Hinweise . . . 328
8.7. Arbeit mit dem Paketfilter . . . 329
8.7.1. Hinzufügen von eigenen Chains und Regeln . . . 329
8.7.2. Einordnen in bestehende Regeln . . . 330
8.7.3. Erweiterung der Paketfilter-Matches . . . 330
8.8. CGI-Erstellung für das httpd-Paket . . . 331
8.8.1. Allgemeines zum Webserver . . . 331
8.8.2. Scriptnamen . . . 331
8.8.3. Menü-Einträge . . . 331
8.8.4. Aufbau eines CGI-Skriptes . . . 332
8.8.5. Benutzerrechte . . . 336
8.8.6. Sonstiges . . . 336
8.8.7. Fehlersuche . . . 336
8.9. Booten, Rebooten, Einwählen und Auflegen unter fli4l . . . 337
8.9.1. Bootkonzept . . . 337
8.9.2. Start-Scripts . . . 337
8.9.3. ttyI devices . . . 340
8.9.4. Scripte beim Einwählen und Auflegen . . . 341
8.10. Paket TEMPLATE . . . 342
8.11. Aufbau der Boot-Diskette . . . 342
8.12. Konfigurationsdateien . . . 343
8.12.1. Konfiguration Provider . . . 343
8.12.2. Konfiguration DNS . . . 343
8.12.3. Hosts-Datei . . . 344
8.12.4. imond-Konfiguration . . . 344

A. Anhang zum Basispaket **345**
A.1. fli4l-Diskette auf Darwin oder Apple Mac OS X erzeugen . . . 345
A.1.1. Voraussetzungen . . . 345
A.1.2. Einleitung . . . 345
A.1.3. fli4l-Distribution entpacken . . . 345
A.1.4. mtools manuell kompilieren . . . 345

A.1.5. Diskette erzeugen . . . 346
A.2. Tokenring – Auszug aus Configure.help des Linux-Kernels . . . 346
A.3. Nullmodemkabel . . . 349
A.4. Serielle Console . . . 350
A.5. Programme . . . 350
A.6. Andere i4l-Tools . . . 351
A.7. Fehlersuche . . . 351
A.8. Literaturhinweise . . . 352
A.9. Präfixe . . . 352
A.10. Gewähr und Haftung . . . 352
A.11. Danke . . . 352
A.11.1. Projektgründung . . . 352
A.11.2. Entwickler- und Testteam . . . 352
A.11.3. Entwickler- und Testteam (nicht mehr aktive) . . . 354
A.11.4. Sponsoren . . . 354
A.12. Feedback . . . 356

B. Anhänge der optionalen Pakete **357**
B.1. CHRONY - Benachrichtigung anderer Applikationen über Timewarps . . . 357
B.2. DSL - PPPD und Active Filter . . . 357
B.3. DYNDNS . . . 358
B.3.1. Hinzufügen von neuen Providern . . . 358
B.3.2. Dank . . . 360
B.3.3. Lizenz . . . 362
B.4. EASYCRON - Crontab in der Boot-Phase ergänzen . . . 362
B.5. HD - Fehler im Zusammenhang mit Festplatten/CompactFlashs . . . 363
B.6. HTTPD . . . 365
B.6.1. Zusätzliche Einstellungen . . . 365
B.6.2. Allgemeine Bemerkungen . . . 366
B.7. IPV6 - Anbindung ans IPv6-Internet mit Hilfe eines SixXS-Tunnels . . . 366
B.7.1. Account erstellen . . . 366
B.7.2. Tunnel konfigurieren . . . 366
B.7.3. Subnetz konfigurieren . . . 369
B.8. ISDN . . . 372
B.8.1. Technische Details zu Einwahl und Routing bei ISDN . . . 372
B.8.2. Fehlermeldungen des ISDN-Subsystems (englisch, i4l-Dokumentation) . . . 374
B.9. UMTS . . . 375
B.9.1. Unterstützte Hardware . . . 375
B.9.2. Modemschnittstelle nicht aktiviert . . . 376

C. Versionsunterschiede **378**
C.1. Unterschiede Version 3.6.2 und 3.6.1 . . . 378
C.2. Unterschiede Version 3.6.2 und 3.6.0 . . . 378
C.3. Unterschiede Version 3.6.2 und 3.4.0 . . . 378

Abbildungsverzeichnis **383**

Tabellenverzeichnis **384**

Index **385**

1. Einleitung

fli4l ist ein Linux-basierender ISDN-, DSL- und Ethernet-Router, der lediglich eine Diskette zum Arbeiten benötigt. Ein 486er mit 16MiB RAM ist dafür vollkommen ausreichend.

Die notwendige Bootdiskette kann unter Unix, Linux oder Windows erstellt werden. Dabei sind keine Linux-Kenntnisse erforderlich, aber evtl. nützlich. Grundkenntnisse von Netzwerken, TCP/IP, DNS und Routing sollten jedoch vorhanden sein. Für eigene Erweiterungen/Entwicklungen, welche über die Standardkonfiguration hinausgehen, sind ein lauffähiges Linux-System und Unix/Linux-Kenntnisse notwendig.

Seit der Version 2.0 unterstützt fli4l auch andere Bootmedien als nur Disketten. Damit ist es möglich, mehr Software-Pakete auf dem fli4l-Router zu installieren, als auf eine Diskette passen. Bevor man sich aber an eine Installation auf Festplatte oder Compact-Flash heranwagt, sollte man vorher einmal die Diskettenversion als Basis-Installation konfigurieren und ausprobieren. Die meisten optionalen Softwarepakete laufen bereits auch bei der Installation auf einer Diskette, so dass ein Massenspeicher (wie Festplatte) nur dann sinnvoll wird, wenn die komplette für fli4l verfügbare Software-Palette installiert werden soll. Es ist auch möglich fli4l von einer CD zu starten.

- Features
  - Erstellen der Boot-Diskette unter Unix, Linux (Seite 262), Darwin/Apple Mac OS X (Seite 345) und Windows (Seite 265)
  - Konfiguration über normale ASCII-Dateien
  - Unterstützung von IP-Masquerading und Port-Forwarding
  - Least-Cost-Routing: automatische Auswahl des Providers, je nach Uhrzeit
  - Anzeige/Berechnung/Protokollierung von Verbindungszeiten und -kosten
  - Windows/Unix/Linux-Client imonc mit Schnittstelle zu imond und telmond
  - Upload von neuen Konfigurationsdateien über Windows-Client imonc
  - Bootdiskette mit vfat-Dateisystem zum dauerhaften Speichern von Dateien
  - Unterstützung von 1680 KiB-Disketten
  - Paketfilter: Logging bei Zugriff von außen auf gesperrte Ports
  - Einheitliche Abbildung von WAN-Schnittstellen auf sogenannte Circuits
  - Betrieb/Routing von ISDN- und DSL-Circuits parallel möglich
- Router
  - Linux-Kernel 2.6.32 (2.6.16 ist auch verfügbar zur besseren Unterstützung älterer Hardware, insbesondere von bestimmten AVM-ISDN/DSL-Adaptern)
  - Paketfilter und IP-Masquerading

  - DNS-Server, damit nicht jede Anfrage von Windows-PCs ins WAN geht
  - Netzwerkfähiger imond-Server mit Monitor-/und LCR-Steuerfunktionen
  - Netzwerkfähiger telmond-Server zur Ausgabe von eingehenden Telefonanrufen
- Ethernet-Support
  - Aktuelle Netzwerkkartentreiber: Unterstützung von über 40 Kartenfamilien
- DSL-Support
  - Roaring Penguin PPPoE-Treiber, mit Dial-on-Demand (abschaltbar)
  - PPTP für DSL-Anbindungen in Österreich und den Niederlanden (EXPERIMENTELL)
- ISDN-Support
  - Aktuelle HiSaX-Treiber: Unterstützung von 37 ISDN-Kartentypen
  - Mehrere ISDN-Verbindungsmöglichkeiten: in/out/callback, raw-ip/ppp
  - Kanalbündelung: automatische Bandbreitenanpassung oder manuelle Zuschaltung des 2. Kanals über Windows-/Unix-Client
- Optionale Programmpakete
  - DNS-Server
  - DHCP-Server
  - SSH-Login
  - Einfache Online-/Offline-Anzeige über LED
  - LCD-Anzeige-Programm mit konfigurierbarem Ausgabeformat
  - Serielle Konsole
  - Mini-Webserver für ISDN- und DSL-Monitoring
  - Zugangserlaubnis für bestimmte konfigurierte Netzwerke von außen
  - Unterstützung für PCMCIA-Karten (heutzutage PC-Karten genannt)
  - Protokollierung von Systemmeldungen: syslogd und klogd
  - Konfiguration von ISAPnP-Karten: isapnp-Werkzeuge
  - Zusätzliche Werkzeuge zum Debugging
  - Konfiguration der seriellen Schnittstelle
  - Notfallsystem zur Fernwartung über ISDN
  - LCD-Display als Monitor: Anzeige von Verbindungen und Übertragungsraten
  - PPP-Server/Router über serielle Schnittstelle
  - ISDN-Modem-Emulator über serielle Schnittstelle
  - Druckerserver
  - Zugriff auf Time-Server zur Synchronisierung der Uhrzeit im Netz

  - Ausführen von Kommandos/Prozeduren bei Telefonanruf (z.B. Internet-Einwahl)
  - Unterstützung von IP-Aliasing (mehrere IPs pro Netzwerkschnittstelle)
  - VPN-Unterstützung
  - IPv6-Unterstützung

- Hardwarevoraussetzungen
  - ISDN: 486er CPU ab 25 MHz, besser 486er ab 33 MHz
  - DSL: 486er CPU ab DX2/66, besser 486er DX4/100 oder Pentium ab 75 MHz
  - 12 MiB Speicher, besser 16 MiB
  - Ethernet-Netzwerkkarte (Unterstützung von über 40 verschiedenen Typen-Familien)
  - ISDN: HiSax-unterstützte ISDN-Karte (Typ 1-39), CAPI-unterstützte ISDN-Karte (13 AVM-Typen) oder ICN 2B aktive ISDN-Karte
  - ein 1440 KiB Diskettenlaufwerk
  - eine Boot-Diskette, alles notwendige drauf
  - optional eine IDE-Festplatte oder ein Compact-Flash-Medium (welches genauso wie eine IDE-Festplatte angesprochen wird) oder ein zweites Diskettenlaufwerk. Alternativ ist auch der Start von CD möglich.

- Softwarevoraussetzungen

  Unter Linux / Unix werden folgende Programme vorausgesetzt:

  - gcc und make
  - syslinux
  - mtools (mcopy)

  Unter Windows werden keine zusätzlichen Werkzeuge benötigt, fli4l bringt alles notwendige mit.

Zusätzlich gibt es zur Steuerung/Statusanzeige des fli4l-Routers noch den Client imonc. Dieses Programm ist für Windows (windows/imonc.exe) und auch für Linux (unix/gtk-imonc) vorhanden.
Und nun ...

Viel Spaß mit fli4l!

Frank Meyer E-Mail: `frank@fli4l.de`

2. Installation und Konfiguration

2.1. Entpacken der Archive

Unter Linux:

```
tar xvfz fli4l-3.6.2.tar.gz
```

Unter Unix:

```
gtar xvfz fli4l-3.6.2.tar.gz
```

Ist gtar nicht vorhanden, geht's auch so:

```
gzip -d < fli4l-3.6.2.tar.gz | tar xvf -
```

Wer die aktuelle Version in einem bereits existierenden fli4l-Verzeichnis installiert, sollte anschließend das Skript mkfli4l.sh -c aufrufen, also:

```
cd fli4l-3.6.2
sh mkfli4l.sh -c
```

Es wird jedoch empfohlen, ein neues Verzeichnis für eine neue Version zu benutzen – die Konfiguration kann durch ein entsprechendes Werkzeug zum Dateivergleich sehr einfach übernommen werden.

Unter Windows kann das komprimierte tar-Archiv zum Beispiel mit WinZip extrahiert werden. Dabei ist jedoch zu beachten, dass die Dateien *mit* Unterverzeichnissen (Einstellung in WinZip überprüfen!) ausgepackt werden. Außerdem ist in *Optionen* ⇒*Konfiguration* die so genannte "Smart TAR CR conversion" abzuschalten. Ist diese eingeschaltet, werden einige (aber wichtige) Dateien von WinZip falsch extrahiert.

Alternativ ist das OpenSource-Programm 7-Zip (`http://www.7-zip.org/`) sehr zu empfehlen, welches ebenso mächtig wie WinZip ist.

Es werden folgende Dateien im Unterverzeichnis `fli4l-3.6.2/` installiert:

- Dokumentation:
  - doc/deutsch/* Deutsche Dokumentation
  - doc/english/* Englische Dokumentation
  - doc/french/* Französische Dokumentation
- Konfiguration:
  - config/*.txt Konfigurationsdateien, diese müssen bearbeitet werden
- Skripte/Prozeduren:

  - mkfli4l.sh Boot-Diskette oder Dateien erzeugen: Linux/Unix-Version
  - mkfli4l.bat Boot-Diskette erzeugen: Windows-Version

- Kernel/Root-Dateisystem:
  - img/kernel Linux-Kernel 2.6.16 oder 2.6.32
  - img/rootfs_distrib.tar Dieses Archiv enthält die Dateien, die nach dem Booten des Kernels als allererstes benötigt werden.

- Zusatzpakete:
  - opt/*.txt Diese Dateien beschreiben, was bei welchen Einstellungen in das Archiv opt.img gelangt.
  - opt/etc/* Standard-Konfigurationsdateien für viele Programme (müssen normalerweise nicht bearbeitet werden).
  - opt/files/* Optionale Kernel-Module, Dateien und Programme

- Quellcode:
  - src/* Quellcode/Werkzeuge für Linux, siehe src/README

- Programme:
  - unix/mkfli4l* Erzeugen der Boot-Disk: Unix/Linux-Version
  - windows/* Erzeugen der Boot-Disk: Windows-Version
  - unix/imonc* imond-Client für Unix/Linux
  - windows/imonc/* imond-Client für Windows

2.2. Konfiguration

2.2.1. Editieren der Konfigurationsdateien

Zur Konfiguration von fli4l müssen lediglich die Dateien config/*.txt angepasst werden. Um im Nachhinein die eigenen Konfiguration mit der ausgelieferten vergleichen zu können oder um mehrere Konfigurationen verwalten zu können, empfiehlt es sich, eine Kopie des config-Verzeichnisses anzulegen und die Konfiguration in dieser Kopie durchzuführen. Ein Vergleich der Konfigurationen ist dann durch Verwendung eines geeigneten Werkzeugs relativ einfach möglich, z.B. unter *nix wie folgt:

```
~/src/fli4l> diff -u {config,dsl-floppy}/build/full_rc.cfg | grep '^[+-]'
--- config/build/full_rc.cfg    2007-03-22 15:34:39.085103706 +0100
+++ dsl-floppy/build/full_rc.cfg        2007-03-22 15:34:31.094317441 +0100
-PASSWORD='/P6h4iOIN5Bbc'
+PASSWORD='3P8F3KbjYgzUc'
-NET_DRV_1='ne2k-pci'
+NET_DRV_1='pcnet32'
-START_IMOND='no'
+START_IMOND='yes'
-OPT_PPPOE='no'
+OPT_PPPOE='yes'
-PPPOE_USER='anonymer'
```

```
-PPPOE_PASS='surfer'
+PPPOE_USER='ich'
+PPPOE_PASS='mein-passwd'
-OPT_SSHD='no'
+OPT_SSHD='yes'
```

Man sieht hier auch sehr schön, dass ein einfacher DSL-Router mit wenigen Handgriffen konfiguriert ist, auch wenn einen die Konfigurationsdateien auf den ersten Blick mit ihrer Fülle von Einstellungsmöglichkeiten erschlagen.

2.2.2. Konfiguration über eine spezielle Konfigurationsdatei

Da sich die Konfiguration durch das Modul-Konzept auf verschiedene Dateien verteilt, und das Bearbeiten dadurch unter Umständen etwas mühsam wird, kann man die Konfiguration auch in einer einzelnen Datei namens *<config verzeichnis>/_fli4l.txt* ablegen, deren Inhalt dann zusätzlich zu den normalen Konfigurationsdateien eingelesen wird und deren Inhalt dominiert. Um beim obigen Beispiel zu bleiben: Um einen einfachen DSL-Router zu konfigurieren, könnten wir einfach folgendes in diese Datei schreiben:

```
PASSWORD='3P8F3KbjYgzUc'
NET_DRV_N='1'
NET_DRV_1='pcnet32'
START_IMOND='yes'
OPT_PPPOE='yes'
PPPOE_USER='ich'
PPPOE_PASS='mein-passwd'
OPT_SSHD='yes'
```

Man sollte vermeiden, beide Konfigurationsvarianten zu mischen.

2.3. Installationsvarianten

In den vorhergehenden Versionen von fli4l wurde lediglich das Booten von einer Diskette unterstützt. Im Normalfall, wenn man fli4l tatsächlich nur als Router verwendet, reicht dies auch vollkommen aus.

Aufgrund der stark angewachsenen Anzahl von möglichen (aber nicht unbedingt notwendigen!) Softwarepaketen für fli4l ist das Booten nunmehr von einer Vielzahl von Bootmedien (Floppy, CD, HD, Netzwerk, Compact-Flash, DoC, ...) möglich und fli4l kann auch auf diversen Medien installiert (HD, Compact-Flash, DoC) werden. Dazu kann fli4l auf 3 verschiedenen Wegen gebootet werden:

Single Image Der Bootloader lädt den Linux-Kern und dann fli4l als ein einziges Image — danach kann fli4l ohne weiteren Zugriff auf andere Medien booten. Beispiele dafür sind die Boottypen *integrated*, *attached*, *netboot* und *cd*.

Split Image Der Bootloader lädt den Linux-Kern und dann ein rudimentäres fli4l-Image, dass die Bootmedien einbindet und die Konfiguration und restlichen Dateien aus einem dort liegenden Archiv holt. Beispiele dafür sind die diversen Floppy-Boottypen, *hd (Typ A)*, *ls120*, *attached* und *cd-emul*.

Installation auf einem Medium Der Bootloader lädt den Linux-Kern und dann ein rudimentäres fli4l-Image, das eine bereits vorhandene fli4l-Installation in sein Dateisystem einbindet und damit keine weiteren Archive auspacken muss. Eine HD-Installation vom Typ B ist ein Beispiel dafür.

Man sollte jedoch zunächst erst einmal fli4l in der 1-Disketten-Version installieren und damit Erfahrungen sammeln. Natürlich ist die Kapazität einer Diskette beschränkt. Möchte man fli4l zusätzlich als Anrufbeantworter und als HTTP-Proxy einsetzen, stößt man schnell an die Kapazitätsgrenzen einer einzigen Diskette. Für diesen Fall nun ist das Booten von Medien ohne die Größenbeschränkung einer Floppy oder die Installation von fli4l auf einer Festplatte oder eines Compact-Flash-Moduls vorgesehen. Jedoch ist die Installation auf Fesplatte etwas komplizierter, so dass man sich erst als fortgeschrittener fli4l-User daran wagen sollte. Wie gesagt: zum Betrieb als Router reicht auch weiterhin eine Diskette, Festplatten sind nicht erforderlich!

Für die Installation ergeben sich daraus die folgenden drei Varianten:

Diskettenrouter Router auf nur einer Diskette, CD, Netzwerk – Die bisherige Installation

HD-Installation Typ A Router auf Festplatte, CF, DoC – nur eine FAT-Partition

HD-Installation Typ B Router auf Festplatte, CF, DoC – je eine FAT- und ext3-Partition

2.3.1. Router auf nur einer Diskette, CD oder vom Netzwerk

Alle benötigten Dateien liegen auf dem Bootmedium und werden beim Booten in eine dynamische RAM-Disk entpackt. In einer Minimalkonfiguration ist damit ein Betrieb des Routers mit nur 8 MiB RAM möglich. Die maximale Konfiguration wird nur durch die Kapazität des Bootmediums und des Hauptspeichers limitiert.

2.3.2. Typ A: Router auf Festplatte – nur eine FAT-Partition

Dies entspricht der Diskettenversion, nur dass die Dateien hierbei auf einer Festplatte liegen, wobei der Begriff „Festplatte“ hier auch Compact-Flash-Medien ab 2 MiB und andere Geräte, welche Linux als Festplatte ansprechen kann, mit einschließt. Seit fli4l 2.1.4 können auch DiskOnChip Flash-Speicher von M-Sys oder SCSI-Festplatten benutzt werden.

Die Beschränkung des Archivs opt.img durch die Diskettenkapazität wird aufgehoben, aber alle diese Dateien müssen in einer RAM-Disk mit der entsprechenden Größe beim Boot installiert werden. Dies erhöht den RAM-Bedarf beim Einsatz vieler Pakete.

Für ein Update der Softwarepakete (d.h. des Archivs opt.img und der rc.cfg über das Netzwerk) muss die FAT-Partition genügend Platz für den Kernel, das RootFS und die DOPPELTE Größe des opt.img haben! Falls auch die Notfall-Option genutzt werden soll, erhöht sich der Platzbedarf noch einmal um die Größe des opt.img.

2.3.3. Typ B: Router auf Festplatte – je eine FAT- und ext3-Partition

Im Gegensatz zum Typ A werden hier nicht alle Dateien in die Ramdisk gepackt, sondern bei dem erstmaligen Start nach der Installation oder nach einem Update aus dem Archiv opt.img direkt auf eine ext3-Partition kopiert und im späteren Betrieb von dort geladen. Bei dieser

Version ist der Speicherbedarf für die RAM-Disk am geringsten und damit meist auch ein Betrieb mit sehr wenig RAM möglich.

Weitere Informationen zur Installation auf Festplatten finden Sie in der Dokumentation des separat herunterzuladenden Pakets HD - beginnend bei der Beschreibung der Variablen `OPT_HDINSTALL`.

3. Basiskonfiguration

Ab Version 2.0 ist die fli4l-Distribution modular aufgebaut und in mehrere Pakete aufgeteilt, die extra heruntergeladen werden müssen. Im Paket `fli4l-3.6.2.tar.gz` ist lediglich die Basis-Software für einen Ethernet-Router enthalten. Für DSL, ISDN und weitere Software müssen die Pakete separat heruntergeladen werden und ausgehend vom Verzeichnis `fli4l-3.6.2/` (!) installiert werden. Durch die Auswahlmöglichkeit des Betriebssystemkerns von fli4l ist dieser in den Kernel Paketen ausgelagert worden. Somit ist als Minimum Basis und ein Kernel Paket erforderlich. In Tabelle 3.1 finden Sie einen Überblick über die Zusatzpakete.

Die zur Konfiguration des fli4l-Routers verwendeten Dateien befinden sich im Verzeichnis `config/` und werden hier im Folgenden beschrieben.

Diese Dateien können mit einem *einfachen* Text-Editor oder auch mit einem speziell an fli4l angepassten Editor verändert werden. Diverse Editoren sind unter
`http://www.fli4l.de/download/zusatzpakete/addons/` zu finden.

Sind spezielle Anpassungen/Erweiterungen erforderlich, die über die unten aufgeführten Einstellungsmöglichkeiten hinausgehen, benötigt man ein lauffähiges Linux-System, um Anpassungen im RootFS vorzunehmen. In diesem Fall hilft `src/README` weiter.

Tabelle 3.1.: Übersicht über die Zusatzpakete

Download-Archiv	Paket
fli4l-3.6.2	BASIS, erforderlich!
kernel_2_6_3x	Kernel 2.6.3x, empfohlen
kernel_2_6_1x	Kernel 2.6.1x, geringer Platzbedarf, enthält alle AVM-Treiber
kernel_2_6_1x_xen	Xen-Kernel 2.6.1x, optional, erforderlich bei Xen Nutzung
fli4l-3.6.2-doc	Komplette Dokumentation
advanced_networking	Erweiterte Netzwerkkonfiguration
chrony	Time-Server/Client
dhcp_client	Verschiedene DHCP-Clients
dns_dhcp	DNS- und DHCP-Server
dsl	DSL-Router (PPPoE, PPTP)
dyndns	Unterstützung von DYNDNS-Diensten
easycron	Zeitplandienst
hd	Installation auf Festplatte
httpd	Mini-Webserver für Status-Ausgaben
imonc_windows	Der Windows-Imonc
imonc_unix	Der GTK-Unix-Imonc
ipv6	Internet Protokoll Version 6
isdn	ISDN-Router
lcd	LCD-Treiber-Software + Statusanzeige
lpdsrv	Druckerserver (ohne Spooling)
openvpn	VPN-Unterstützung
pcmcia	Unterstützung von PCMCIA-Karten
ppp	PPP-Anbindung über serielle Schnittstelle
proxy	Proxy-Server
qos	Quality of Service
sshd	SSH-Server
tools	Diverse Linux-Werkzeuge
umts	Anbindung mittels UMTS an das Internet
usb	Unterstützung der USB-Schnittstelle
wlan	Unterstützung von WLAN-Karten

3.1. Beispiel-Datei

Die Beispiel-Datei `base.txt` im Verzeichnis `config/` hat folgenden Inhalt:

```
##-------------------------------------------------------------------------
## base.txt - fli4l configuration parameters                      __FLI4LVER__
##
## You can edit/change this file with any text editor.
##
##       P L E A S E   R E A D   T H E   D O C U M E N T A T I O N ,
##                     S E E   R E A D M E . T X T
##
```

```
##       B I T T E  U N B E D I N G T  D I E  D O K U M E N T A T I O N
##            L E S E N ,  S I E H E  R E A D M E . T X T
##
## Creation:     26.06.2001  fm
## Last Update:  $Id: base.txt 22941 2012-07-23 20:34:57Z kristov $
##
## Copyright (c) 2001-2011 - Frank Meyer, fli4l-Team - team@fli4l.de
##
## This program is free software; you can redistribute it and/or modify
## it under the terms of the GNU General Public License as published by
## the Free Software Foundation; either version 2 of the License, or
## (at your option) any later version.
##-----------------------------------------------------------------------------

#-----------------------------------------------------------------------------
# General settings:
#-----------------------------------------------------------------------------
HOSTNAME='fli4l'               # name of fli4l router
PASSWORD='fli4l'               # password for root login (console, sshd, imond)
BOOT_TYPE='fd'                 # boot device: fd, fdx2, dualfd, hd, cd, ls120,
                               # integrated, attached, netboot, pxeboot
                               # (cd, cdemul, hd and ls120 need an
                               # installed opt hd)
MOUNT_BOOT='rw'                # mount boot device (floppy): ro, rw, no
BOOTMENU_TIME='5'              # waiting time of bootmenu in seconds
                               # to activate normal boot
TIME_INFO='MEZ-1MESZ,M3.5.0,M10.5.0/3'
                               # description of local time zone,
                               # don't touch without reading documentation
KERNEL_VERSION='2.6.32.59'     # kernel version
KERNEL_BOOT_OPTION=''          # append option to kernel command line
COMP_TYPE_KERNEL='lzma'        # compression-type for kernel:
                               # gzip, bzip2, lzma,  uncompressed
COMP_TYPE_ROOTFS='lzma'        # compression-type for rootfs: gzip, bzip2, lzma
COMP_TYPE_OPT='lzma'           # compression-type for opt: bzip2, lzma
IP_CONNTRACK_MAX=''            # override maximum limit of connection
                               # tracking entries
POWERMANAGEMENT='none'         # select pm interface: none, acpi, apm, apm_rm
                               # apm_rm switches to real mode before invoking
                               #   apm power off

#-----------------------------------------------------------------------------
# Localisation
#-----------------------------------------------------------------------------
LOCALE='de'                    # defines the default language for several
                               # components, such as httpd

#-----------------------------------------------------------------------------
# Mount extra floppy drive to /floppy:
#-----------------------------------------------------------------------------
OPT_MOUNTFLOPPY='no'           # mount extra floppy drive: first, second, no

#-----------------------------------------------------------------------------
```

```
# Console settings (serial console, blank time, beep):
#-------------------------------------------------------------------------
CONSOLE_BLANK_TIME=''          # time in minutes (1-60) to blank
                               # console; '0' = never, '' = system default
BEEP='yes'                     # enable beep after boot and shutdown
SER_CONSOLE='no'               # use serial interface instead of or as
                               # additional output device and main input device
    SER_CONSOLE_IF='0'         # serial interface to use, 0 for ttyS0 (COM1)
    SER_CONSOLE_RATE='9600'    # baudrate for serial console

#-------------------------------------------------------------------------
# Debug Settings:
#-------------------------------------------------------------------------
DEBUG_STARTUP='no'             # write an execution trace of the boot

#-------------------------------------------------------------------------
# Keyboard layout
#-------------------------------------------------------------------------
KEYBOARD_LOCALE='auto'         # auto: use most common keyboard layout for
                               # the language specified in 'LOCALE'
OPT_MAKEKBL='no'               # set to 'yes' to make a new local keyboard
                               # layout map on the fli4l-router

#-------------------------------------------------------------------------
# Ethernet card drivers:
#-------------------------------------------------------------------------
#
#  please see file base_nic.list in your config-dir or read the documentation
#
#
#  If you need a dummy device, use 'dummy' as your NET_DRV
#  and IP_NET_%_DEV='dummy<number>' as your device
#
#-------------------------------------------------------------------------
NET_DRV_N='1'                  # number of ethernet drivers to load, usually 1
NET_DRV_1='ne2k-pci'           # 1st driver: name (e.g. NE2000 PCI clone)
NET_DRV_1_OPTION=''            # 1st driver: additional option
NET_DRV_2='ne'                 # 2nd driver: name (e.g. NE2000 ISA clone)
NET_DRV_2_OPTION='io=0x320'    # 2nd driver: additional option

#-------------------------------------------------------------------------
# Ether networks used with IP protocol:
#-------------------------------------------------------------------------
IP_NET_N='1'                   # number of IP ethernet networks, usually 1
IP_NET_1='192.168.6.1/24'      # IP address of your n'th ethernet card and
                               # netmask in CIDR (no. of set bits)
IP_NET_1_DEV='eth0'            # required: device name like ethX

#-------------------------------------------------------------------------
# Additional routes, optional
#-------------------------------------------------------------------------
IP_ROUTE_N='0'                             # number of additional routes
IP_ROUTE_1='192.168.7.0/24 192.168.6.99'  # network/netmaskbits gateway
```

```
IP_ROUTE_2='0.0.0.0/0 192.168.6.99'       # example for default-route

#-------------------------------------------------------------------------
# Packetfilter configuration; there are two styles, old and new -
# you have to choose one of them.
#
# Please see documentation if you want to use the old style and add all needed
# config VARS to configuration
#
# New style packet filter config:
#-------------------------------------------------------------------------

PF_NEW_CONFIG='yes'                     # new style packet filter config

    PF_INPUT_POLICY='REJECT'            # be nice and use reject as policy
    PF_INPUT_ACCEPT_DEF='yes'           # use default rule set
    PF_INPUT_LOG='no'                   # don't log anything
    PF_INPUT_LOG_LIMIT='3/minute:5'     # log 3 events per minute; allow a
                                        # burst of 5 events
    PF_INPUT_REJ_LIMIT='1/second:5'     # reject 1 connection per second; allow
                                        # a burst of 5 events; otherwise
                                        # drop packet
    PF_INPUT_UDP_REJ_LIMIT='1/second:5' # reject 1 udp packet per second; allow
                                        # a burst of 5 events; otherwise drop
                                        # packet
    PF_INPUT_N='1'
    PF_INPUT_1='IP_NET_1 ACCEPT'        # allow all hosts in the local
                                        # network access to the router
    PF_INPUT_2='tmpl:samba DROP NOLOG'  # drop (or reject) samba access
    PF_INPUT_2_COMMENT='no samba traffic allowed' # without logging, otherwise
                                                  # the log file will be filled
                                                  # with useless entries

    PF_FORWARD_POLICY='REJECT'          # be nice and use reject as policy
    PF_FORWARD_ACCEPT_DEF='yes'         # use default rule set
    PF_FORWARD_LOG='no'                 # don't log anything
    PF_FORWARD_LOG_LIMIT='3/minute:5'   # log 3 events per minute; allow a
                                        # burst of 5 events
    PF_FORWARD_REJ_LIMIT='1/second:5'   # reject 1 connection per second; allow
                                        # a burst of 5 events; otherwise
                                        # drop packet
    PF_FORWARD_UDP_REJ_LIMIT='1/second:5'   # reject 1 udp packet per second;
                                            # allow a burst of 5 events;
                                            # otherwise drop packet
    PF_FORWARD_N='2'
    PF_FORWARD_1='tmpl:samba DROP'      # drop samba traffic if it tries
                                        # to leave the subnet
    PF_FORWARD_2='IP_NET_1 ACCEPT'      # accept everything else

    PF_POSTROUTING_N='1'
    PF_POSTROUTING_1='IP_NET_1 MASQUERADE'  # masquerade traffic leaving
                                            # the subnet
```

```
    PF_PREROUTING_N='0'
    PF_PREROUTING_1='1.2.3.4 dynamic:22 DNAT:@client2'
                                          # forward ssh connections
                                          # coming from 1.2.3.4 to client2

    PF_USR_CHAIN_N='0'

#-------------------------------------------------------------------------
# Simple DMZ setup for dial-up based routers -- see documentation
#-------------------------------------------------------------------------
OPT_DMZ='no'

#-------------------------------------------------------------------------
# Optional package: PORTFW
#-------------------------------------------------------------------------
PORTFW_N='0'                          # how many portforwardings to set up
PORTFW_1_TARGET='8080'                # example 1: forward ext. port 8080
PORTFW_1_NEW_TARGET='192.168.6.15:80' # ...to int. host 192.168.6.15 port 80
PORTFW_1_PROTOCOL='tcp'               # ...using tcp
PORTFW_2_TARGET='3000-3010'           # example 2: forward portrange 3000-3010
PORTFW_2_NEW_TARGET='192.168.6.15'    # ...to int. host 192.168.6.15
PORTFW_2_PROTOCOL='tcp'               # ...using tcp

#-------------------------------------------------------------------------
# Masq modules
#-------------------------------------------------------------------------
MASQ_MODULE_N='0'                     # load n masq modules (see documentation)
MASQ_MODULE_1='ftp'                   # ftp
MASQ_MODULE_1_OPTION=''               # options, see documentation
MASQ_MODULE_2='irc'                   # irc
MASQ_MODULE_2_OPTION=''               # options, see documentation

#-------------------------------------------------------------------------
# Domain configuration:
# configuration of DNS, DHCP-Server and HOSTS -> see package DNS_DHCP
#-------------------------------------------------------------------------
DOMAIN_NAME='lan.fli4l'               # your domain name
DNS_FORWARDERS='194.8.57.8'           # DNS servers of your provider,
                                      # e.g. ns.n-ix.net

# optional configuration for the host-entry of the router in /etc/hosts
#HOSTNAME_IP='IP_NET_1_IPADDR'        # IP to bind to HOSTNAME
#HOSTNAME_ALIAS_N='0'                 # how many ALIAS names for the router
#HOSTNAME_ALIAS_1='router.lan.fli4l'  # first ALIAS name
#HOSTNAME_ALIAS_2='gateway.my.lan'    # secound ALIAS name

#-------------------------------------------------------------------------
# imond configuration:
#-------------------------------------------------------------------------
START_IMOND='no'                    # start imond: yes or no
    IMOND_USE_ORIG='yes'            # use the original version of imond instead
                                    # of the development version: yes or no
    IMOND_PORT='5000'               # port (tcp), don't open it to the outside
```

```
    IMOND_PASS=''                   # imond-password, may be empty
    IMOND_ADMIN_PASS=''             # imond-admin-password, may be empty
    IMOND_LED=''                    # tty for led: com1 - com4 or empty
    IMOND_BEEP='no'                 # beep if connection going up/down
    IMOND_LOG='no'                  # log /var/log/imond.log: yes or no
    IMOND_LOGDIR='/var/log'         # log-directory, e.g. /var/log
    IMOND_ENABLE='yes'              # accept "enable/disable" commands
    IMOND_DIAL='yes'                # accept "dial/hangup" commands
    IMOND_ROUTE='yes'               # accept "route" command
    IMOND_REBOOT='yes'              # accept "reboot" command

#-------------------------------------------------------------------------
# Generic circuit configuration:
#-------------------------------------------------------------------------
IP_DYN_ADDR='yes'                   # use dyn. IP addresses (most providers do)
DIALMODE='auto'                     # standard dialmode: auto, manual, or off

#-------------------------------------------------------------------------
# optional package: syslogd
#-------------------------------------------------------------------------
OPT_SYSLOGD='no'                        # start syslogd: yes or no
    #SYSLOGD_RECEIVER='no'              # receive messages from network
    SYSLOGD_DEST_N='1'                  # number of destinations
    SYSLOGD_DEST_1='*.* /dev/console'   # n'th prio & destination of syslog msgs
    SYSLOGD_DEST_2='*.* @192.168.6.2'   # example: loghost 192.168.6.2
    SYSLOGD_DEST_3='kern.info /var/log/dial.log'  # example: log infos to file

    SYSLOGD_ROTATE='no'                 # rotate syslog-files once every day
    SYSLOGD_ROTATE_DIR='/data/syslog'   # move rotated files to ....

#-------------------------------------------------------------------------
# Optional package: klogd
#-------------------------------------------------------------------------
OPT_KLOGD='no'                      # start klogd: yes or no

#-------------------------------------------------------------------------
# Optional package: logip
#-------------------------------------------------------------------------
OPT_LOGIP='no'                      # logip: yes or no
    LOGIP_LOGDIR='/boot'            # log-directory, e.g. /boot

#-------------------------------------------------------------------------
# Optional package: y2k correction
#-------------------------------------------------------------------------
OPT_Y2K='no'                        # y2k correction: yes or no
    Y2K_DAYS='0'                    # correct hardware y2k-bug: add x days

#-------------------------------------------------------------------------
# Optional package: PNP
#-------------------------------------------------------------------------
OPT_PNP='no'                        # install isapnp tools: yes or no
```

Zu beachten ist, dass diese Datei im DOS-Format gespeichert ist. Das heißt, sie enthält jeweils

am Zeilenende ein zusätzliches Carriage-Return (CR). Da die meisten Unix-Editoren damit keine Probleme bekommen, habe ich mich für dieses Format entschlossen, denn umgekehrt hat der Windows-Editor bei fehlendem CR am Zeilenende keine Chance!

Sollte es wider Erwarten unter Unix/Linux doch Probleme mit dem Lieblingseditor geben, kann die Datei vor dem Editieren mit einem Befehl in das Unix-Format konvertiert werden:

```
sh unix/dtou config/base.txt
```

Für die Erstellung der Boot-Diskette ist es völlig unerheblich, ob die Datei CRs am Zeilenende enthält oder nicht. Sie werden beim Schreiben auf die Bootdiskette einschließlich der Kommentare komplett ignoriert.

Jetzt aber zum Inhalt ...

3.2. Allgemeine Einstellungen

HOSTNAME Als erstes sollte man seinem fli4l-Router einen Namen geben. Im Beispiel ist dies `fli4l`.

PASSWORD Das hier angegebene Passwort wird für das Einloggen in den fli4l-Rechner benötigt – sei es per Tastatur direkt am Router oder per SSH von einem anderen Rechner aus (hierzu wird das sshd-Paket benötigt).

Das Passwort ist ab Version 1.5 zwingend.

Standard-Einstellung: `PASSWORD='fli4l'`

BOOT_TYPE Standardwert: `BOOT_TYPE='fd'`

`BOOT_TYPE` legt im weitesten Sinne das Bootmedium fest. Diese Variable steuert, welche zusätzlichen Treiber (Kernel-Module) und Start-Skripte mit in das RootFS aufgenommen werden. Zum Verständnis eine kurze Skizze des Bootvorgangs:

- Das BIOS des Rechners lädt/startet den Bootloader auf dem Bootmedium.
- Der Bootloader (i.d.R syslinux) entpackt, lädt und startet den Kernel.
- Der Kernel entpackt das RootFS (= das grundlegende Dateisystem mit darin enthalten Programmen und Skripte), mountet das RootFS und beginnt die Start-Skripte abzuarbeiten.
- Je nach `BOOT_TYPE` werden nun die Kernel-Module für das jeweilige Bootmedium geladen, die Boot-Partition gemountet und das Opt-Archiv (`opt.img`) mit den zusätzlichen Programmen entpackt.
- Im Anschluss beginnt die Konfiguration der einzelnen Dienste des fli4l.

Zur Zeit sind folgende Werte für `BOOT_TYPE` gültig:

fd Der Disketten-Boot von Laufwerk "a:"; die Größe des Bootmediums (1440 KiB oder 1680 KiB) wird automatisch ermittelt. Die Disketten sind hierzu gesondert zu formatieren, näheres ist bitte dem Kapitel Erstellen der Bootdiskette (Seite 272) zu entnehmen.

fdx2 Mit einem Laufwerk "a:" und zwei Disketten booten. Die Disketten müssen beim Booten gewechselt werden. Die formatierte Diskettenkapzität der zweiten Diskette darf nicht kleiner sein als die der ersten Diskette.

dualfd Ein Disketten-Boot von den Laufwerken "a:" und "b:". Die formatierte Diskettenkapzität der zweiten Diskette darf nicht kleiner sein als die der ersten Diskette.

ls120 Boot von LS120/240 sowie ZIP Disks. Hierzu wird das Paket HD und der Treiber ide-floppy benötigt.

hd Boot von Festplatte. Hierzu wird das Paket HD benötigt. Näheres ist der Dokumentation (Seite 125) zum Paket HD zu entnehmen.

cd, cdemul Boot von CD-ROM. Auch hierzu wird das Paket HD benötigt, weil dort die Treiber für CD-Laufwerke drin stecken. Es wird lediglich das ISO-Image fli4l.iso der CD erzeugt, welches anschliessend mit dem jeweiligen Lieblingsbrennprogramm selbst auf CD gebrannt werden muss. Bitte darauf achten, dass hier auch der richtige CD-Treiber ausgewählt wurde, ide-cd ist nur für IDE-Laufwerke, scsi-cd alleine funktioniert nicht, hier ist unbedingt auch noch der passende SCSI-Hostadapter-Treiber auszuwählen. cdemul ist eine Alternative für älteren Boards wo cd nicht funktioniert.

integrated Bei diesem Typ wird kein Bootmedium zu Grunde gelegt, sondern das Opt-Archiv vollständig ins RootFS integriert. Somit entfällt das Mounten des Bootmediums und der Kernel kann gleich alles entpacken. Dieser `BOOT_Type` wird z.B. fürs Booten vom Netzwerk benötigt.
Hinweis: Ein remote Update ist natürlich in diesem Fall nicht möglich.

attached Ähnlich wie **integrated**, jedoch wird das Opt-Archiv als Datei `opt.img` ans RootFS angehängt, mit der Folge, dass es wieder im Verzeichnis `/boot` zu finden ist und gesondert beim Bootvorgang entpackt wird.
Ansonsten gilt das unter `integrated` Gesagte.

netboot Entspricht **integrated**. Es wird jedoch zusätzlich das Skript `mknetboot.sh` gestarted, welches ein Image zum Booten via LAN erzeugt. Weiteres ist bitte dem *Mini-Howto für fli4l und Netzboot* zu entnehmen.
Hinweis: fli4l bringt weder eine Ether- oder Netbootumgebung mit. Diese ist gesondert zu installieren!

PRESERVE Mit dieser **optionalen** Variablen kann man einstellen ob, bei Verwendung von zwei Disketten (`BOOT_TYPE`='fdx2' oder 'dualfd'), auf der zweiten Diskette Platz reserviert werden soll für z.B. DHCP-Leases. Die Angabe wird in Bytes gemacht. Man braucht sich nicht um Cluster zu kümmern, beim Build wird auf Cluster abgerundet.

Beispiel:

```
PRESERVE='40000'
```

Im Moment wird diese Variable nur beim Windows-Build berücksichtigt.

Standardeinstellung: `PRESERVE`='0'

MOUNT_BOOT Hier wird eingestellt, wie das Boot-Medium gemountet werden soll. Es gibt drei Möglichkeiten:

rw – Read/Write – Schreiben und Lesen ist möglich

ro – Read-Only – Nur Lesen ist möglich

no – None – Medium wird nach dem Boot wieder abgemeldet und kann dann bei Bedarf entnommen werden.

Bei bestimmten Konfigurationen ist es unbedingt erforderlich, das Medium Read/Write anzumelden, z.B. wenn man den DHCP-Server einsetzen oder die imond-Log-Datei auf dem Medium anlegen möchte.

Standard-Einstellung: `MOUNT_BOOT`='rw'

BOOTMENU_TIME Hier wird eingestellt, wie lange der syslinux Bootloader warten soll, bis automatisch mit der Standard-Installation gebootet wird.

Im Paket HD besteht die Möglichkeit, über `OPT_RECOVER` eine Funktion zu aktivieren, mit der eine Notfallinstallation aus einer laufenden Installation erstellt werden kann. Diese kann im Bootmenü über die Wahl der Recover-Version aktiviert werden.

Sollte hier der Wert '0' eingestellt sein, wartet der syslinux Bootloader bis der Anwender die Standard- oder die Recover-Version auswählt und aktiviert!

Standard-Einstellung: `BOOTMENU_TIME`='20'

TIME_INFO Uhren ticken in der Unix-Welt und damit auch unter fli4l normalerweise nach der UTC (Coordinated Universal Time), einer weltweit einheitlichen Uhrzeit, die vor der Verwendung in die lokale Zeit umgerechnet wird. `TIME_INFO` liefert fli4l die dafür notwendigen Informationen über die Namen der Zeitzonen, die Differenz zu UTC und Regeln, wann auf Sommerzeit und wieder zurück gewechselt wird. Damit das korrekt funktioniert, muss die Hardware Uhr auf UTC gestellt werden (entspricht der Londoner Winterzeit) oder über das Paket chrony mit einem Timeserver synchronisiert werden (diese liefern UTC aus).

Die Einträge in `TIME_INFO` bedeuten dabei folgendes:

```
TIME_INFO='MEZ-1MESZ,M3.5.0,M10.5.0/3'
```

- *MEZ-1:* Wir befinden uns in der mitteleuropäischen Zeitzone (*MEZ*, die der UTC eine Stunde voraus ist *MEZ-1=UTC*.
- *MESZ:* In dieser Zeitzone gibt es Sommerzeit (Mitteleuropäische Sommerzeit). Da nichts weiter angegeben wird, kommt man zur Sommerzeit, indem man die Zeit eine Stunde vorstellt.
- *M3.5.0,M10.5.0/3:* Am letzten Sonntag im März (um 2 Uhr) wird zur Sommerzeit gewechselt, am letzten Sonntag im Oktober (um 3 Uhr) wieder zurück.

Normalerweise braucht man diesen Wert nie anzufassen, es sei denn man sitzt in einer anderen Zeitzone. Will man die Werte anpassen, sollte man einen Blick auf die Spezifikation der Umgebungsvariable TZ werfen, die unter folgender URL zu finden ist (englisch): http://pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap08.html

KERNEL_VERSION Legt die Version des zu verwendenden Kerns fest. Entsprechend dieser Variable werden der Kern aus *img/kernel-<kernel version>.<compression extension>*

und die Kernel-Module aus *opt/files/lib/modules/<kernel version>* selektiert. Für neuere Kernel ist auch die Extension -smp möglich, die neben der Unterstützung für mehrere CPU und mehr Ram auch Erweiterungen für xen-DomU und kvm beinhaltet. Diese Version setzt jedoch mindestens einen Pentium-Prozessor mit PAE-Support voraus.

KERNEL_BOOT_OPTION Der Inhalt dieser Variable wird an die Kommandozeile des Kerns in der syslinux.cfg angehängt. Für Systeme ohne Tastatur-Controller bietet sich hier 'nokbd' an. Manche Systeme benötigen für korrekten Reboot 'reboot=bios'. Bei WRAP-Systemen also 'nokbd reboot=bios'.

COMP_TYPE_KERNEL Der Inhalt dieser Variable legt den Typ der Kompression für den Kernel fest. Mögliche Werte sind 'gzip', 'lzma' und 'uncompressed'. Letzteres benötigt man für xen, da manche Varianten mit lzma-Kernel nichts anfangen können.

COMP_TYPE_ROOTFS Der Inhalt dieser Variable legt den Typ der Kompression für das RootFS fest. Mögliche Werte sind 'gzip', 'bzip2' und 'lzma'.

COMP_TYPE_OPT Der Inhalt dieser Variable legt den Typ der Kompression für das OPT-Archiv fest. Mögliche Werte sind 'bzip2' und 'lzma'.

POWERMANAGEMENT Der Kern unterstützt verschiedene Formen des Powermanagements, das etwas betagte APM und das aktuellere ACPI. Hier kann man einstellen, welche Form er verwenden soll. Mögliche Werte sind 'none' (kein Powermanagement), 'acpi' und die beiden APM-Varianten 'apm' und 'apm_rm'. Letzteres schaltet in einen speziellen Prozessormodus, bevor der Router ausgeschaltet wird.

IP_CONNTRACK_MAX Mit Hilfe dieser Variable kann man die maximale mögliche Anzahl gleichzeitiger Verbindungen manuell einstellen. Normalerweise wird anhand des eingebauten Arbeitsspeichers automatisch ein sinnvoller Wert ermittelt. In Tabelle 3.2 sind die verwendeten Voreinstellungen zusammengefasst dargestellt.

Tabelle 3.2.: Automatische Einstellung der maximalen Verbindungsanzahl

Arbeitsspeicher in MiB	gleichzeitige Verbindungen
16	1024
24	1280
32	2048
64	4096
128	8192

Bei Einsatz von FileSharing-Programmen hinter oder auf dem Router und wenig Arbeitsspeicher ist die maximale Anzahl gleichzeitiger Verbindungen aber sehr schnell erreicht und zusätzliche Verbindungen können nicht mehr aufgebaut werden.
Das äußert sich in Fehlermeldungen wie

```
ip_conntrack: table full, dropping packet
```

oder

```
ip_conntrack: Maximum limit of XXX entries exceeded
```

Mittels IP_CONNTRACK_MAX lässt sich nun die maximale Anzahl gleichzeitiger Verbindungen fest auf einen bestimmten Wert einstellen. Aber seid gewarnt. Jede einzelne mögliche Verbindung kostet 350 Bytes an Arbeitsspeicher, der nicht mehr für andere Dinge genutzt werden kann. Setzt man also 10000, so sind gerundet 3,34 MB Arbeitsspeicher für den normalen Gebrauch verloren (Kernel, Ramdisks, Programme).

Bei 32 MiB RAM sollte es kein Problem sein, mal eben 2 oder 3 MiB für die ip_conntrack-Tabelle zu reservieren, bei 16 MiB wird es knapp und bei 12 oder sogar 8 MiB ist absolute Sparwut angesagt.

Die momentan benutze Einstellung lässt sich auf der Konsole mit

```
cat /proc/sys/net/ipv4/ip_conntrack_max
```

anzeigen und mit

```
echo "XXX" > /proc/sys/net/ipv4/ip_conntrack_max
```

zur Laufzeit setzen, wobei XXX für die Anzahl der Einträge steht. Die Einträge in der IP_CONNTRACK-Tabelle selbst können auf der Konsole mit

```
cat /proc/net/ip_conntrack
```

angesehen und mit

```
cat /proc/net/ip_conntrack | grep -c use
```

gezählt werden.

Standard-Einstellung: IP_CONNTRACK_MAX=”

LOCALE Standardeinstellung: LOCALE=’de’

Einige Komponenten sind mittlerweile mehrsprachfähig. Dazu zählen beispielsweise das Konsolen-Menü und die Weboberfläche. Mit dieser Variablen können Sie die bevorzugte Sprache auswählen. Verschiedene Komponenten haben noch eine eigene Einstellung womit diese Grundeinstellung, wenn nötig, überlistet werden kann. Wenn die hier angegebene Sprache bei einem Komponenten (noch) nicht verfügbar ist, wird auf Englisch zurückgefallen.

Wenn KEYBOARD_LOCALE=’auto’ wird versucht ein zu der LOCALE-Einstellung passendes Tastatur-Layout ein zu stellen.

Bisher sind folgende Einstellungen möglich: de, en, es, fr, hu, nl.

3.3. Einbinden von zusätzlicher Diskette

OPT_MOUNTFLOPPY In einigen Konfiguration (z.B. bei Boot von CD oder Netzwerk) ist das Einbinden einer Diskette bzw. einer zusätzlichen Diskette um z.B. die Log-Dateien oder die DHCP-Lease-Datei zu speichern sinnvoll. Mit OPT_MOUNTFLOPPY=’first’ wird das erste Diskettenlaufwerk (a:) unter dem Verzeichnis /floppy in das Dateiensystem eingebunden, mit OPT_MOUNTFLOPPY=’second’ das zweite Laufwerk (b:). Mit OPT_MOUNTFLOPPY=’no’ wird das Einbinden deaktiviert.

3.4. Konsolen-Einstellungen

CONSOLE_BLANK_TIME Normalerweise aktiviert der Linux-Kern nach einer gewissen Zeit ohne Eingaben auf dem aktuellen Eingabegerät den Bildschirmschoner. Mit der Variable `CONSOLE_BLANK_TIME` kann man diese Zeit konfigurierereren bzw. den Bildschirmschonermodus ganz deaktivieren (`CONSOLE_BLANK_TIME`='0').

BEEP Signalton am Ende des Boot- und Shutdownprozesses ausgeben.

Trägt man hier 'yes' ein, ertönt ein Signalton am Ende des Boot- und Shutdownprozesses. Wer extremen Platzmangel auf der Diskette hat, oder keinen Signalton möchte, kann hier also 'no' eingetragen lassen.

SER_CONSOLE Fli4l kann gänzlich ohne Tastatur und Grafikkarte eingesetzt werden. Damit man den Router auch ohne Telnet oder SSH bedienen sowie alle Boot-Meldungen sehen kann, ist es möglich, die Ein- und Ausgaben auf die serielle Schnittstelle umzuleiten. Dazu muss `SER_CONSOLE`, `SER_CONSOLE_IF` sowie `SER_CONSOLE_RATE` angepasst werden.

Die serielle Konsole kann in drei Modi betrieben werden:

`SER_CONSOLE`	Konsolen-Ein-/Ausgabe
no	Ein- und Ausgabe über tty0
yes	Ein- und Ausgabe über serielle Konsole
primary	Ein- und Ausgabe über serielle Konsole, Ausgabe zusätzlich über tty0
secondary	Ein- und Ausgabe über tty0, Ausgabe zusätzlich über serielle Konsole

Standard-Einstellung: `SER_CONSOLE`='no'

Wenn der Wert von `SER_CONSOLE` verändert wird, wird diese Änderung nur bei der Erstellung einer neue Diskette oder beim remote-Update der syslinux.cfg wirksam.

Achten Sie beim Ausschalten der Seriellen Konsole darauf, sich einen alternativen Zugang zum Router (ssh oder direkt an der Tastatur) zu erhalten.

Weitere Informationen sind im Anhang unter Serielle Konsole (Seite 350) zu finden.

SER_CONSOLE_IF Nummer der seriellen Schnittstelle für die Konsolenausgabe.

Hier ist die Nummer der Schnittstelle einzutragen, an die die serielle Konsole angeschlossen wird. 0 entspricht ttyS0 bzw. COM1 in der DOS-Welt.

SER_CONSOLE_RATE Übertragungsrate der seriellen Schnittstelle für die Konsolenausgabe.

Hier trägt man die Baud-Rate ein, mit der die Daten auf der seriellen Schnittstelle übertragen werden. Sinnvolle Werte: 4800, 9600, 19200, 38400, 57600, 115200

3.5. Loggen der Bootsequenz und des Ladens von Modulen

Fli4l loggt die gesamten Ausgaben des Bootvorganges in einer Datei (*/var/tmp/boot.log*). Diese Datei kann man sich am Ende des Bootvorganges auf der Konsole ansehen oder über den entsprechenden Menüpunkt im Web-Interface.

Manchmal ist es jedoch sinnvoll, bei Problemen einen ausführlicheren Trace der Start-Sequenz zu generieren um den Bootvorgang hinterher auf Probleme untersuchen zu können. Dazu dient `DEBUG_STARTUP`.

DEBUG_STARTUP Steht dieser Wert auf 'yes', wird beim Booten jedes ausgefuehrte Kommando vor seiner Ausführung auf den Schirm geschrieben. Da für das korrekte Funktionieren Änderungen an der syslinux.cfg vorgenommen werden müssen, gilt das für `SER_CONSOLE` Gesagte auch hier. Wenn man die syslinux.cfg von Hand ergänzen will, ist es nötig, ein `fli4ldebug=yes` einzufügen. `DEBUG_STARTUP` muss dann aber trotzdem auf 'yes' stehen.

DEBUG_MODULES Einige Module werden automatisch vom Kern geladen, ohne dass man das vorher erkennen kann. `DEBUG_MODULES='yes'` aktiviert einen Modus, der einem die kompletten Modulladesequenzen zeigt, egal, ob sie von einem Skript oder vom Kern angestoßen werden.

3.6. Verwendung einer eigenen /etc/inittab

Man kann von Init zusätzliche Programme auf zusätzlichen Konsolen starten lassen oder die Standardkommandos der Init-Konfigurationsdatei verändern. Ein Eintrag sieht wie folgt aus:

```
device:runlevel:action:command
```

Das *device* ist das Gerät, über das das Programm seine Ein-/Ausgaben machen soll. Möglich sind hier die normalen Terminals tty1-tty4 oder serielle Terminals ttyS0-ttySn mit $n <$ Anzahl der vorhandenen seriellen Schnittstellen.

Als *action* kommen in der Regel *askfirst* oder *respawn* in Frage. Bei askfirst wird auf einen tastendruck gewartet, bevor das Kommando ausgeführt wird, bei respawn wird es automatisch neu gestartet, wenn sich das Programm beendet.

command ist das Programm, das ausgeführt werden soll. Es ist mit vollständiger Pfadangabe zu spezifizieren.

Die Busybox-Dokumentation unter http://www.busybox.net enthält eine genaue Beschreibung des inittab Formats.

Die normale inittab sieht wie folgt aus:

```
::sysinit:/etc/rc
::respawn:/usr/local/bin/mini-login
::ctrlaltdel:/sbin/reboot
::shutdown:/etc/rc0
::restart:/sbin/init
```

Diese könnte man z.B. um den Eintrag

```
tty2::askfirst:/usr/local/bin/mini-login
```

erweitern, um ein zweites Login auf Terminal zwei zu bekommen. Dazu einfach die Datei opt/etc/inittab nehmen, nach <config verzeichnis>/etc/inittab kopieren und mit einem Editor bearbeiten.

3.7. Länderspezifische Tastaturlayouts

KEYBOARD_LOCALE Wenn man öfter direkt auf dem fli4l Router arbeitet ist ein lokales Tastaturlayout eine willkommene Hilfe. Mit `KEYBOARD_LOCALE='auto'` wird versucht, ein Tastaturlayout passend zu der Einstellung von `LOCALE` zu benutzen. Mit der Einstellung `"` wird kein lokales Tastaturlayout auf dem fli4l–Router installiert; es wird dann das im Kernel anwesende Standardlayout verwendet. Alternativ kann man auch den Namen einer lokalen Tastaturlayoutmap direkt angeben. Wenn z.B. `'de-latin1'` eingestellt wird, prüft der Buildprozess ob in opt/etc eine Datei mit Namen de-latin1.map vorliegt. Wenn ja, wird die entsprechende .map-Datei als Tastaturlayout eingebunden.

OPT_MAKEKBL Wenn man für seine Tastatur eine map Datei erstellen will muss man wie folgt vorgehen:

- `OPT_MAKEKBL` auf 'yes' setzen.
- Auf dem Router 'makekbl.sh' anrufen. Vorzugsweise macht man dies über eine ssh-Verbindung weil das Tastaturlayout sich ändert und das lästig sein kann.
- Die Anweisungen befolgen.
- Ihre neue <locale>.map Datei liegt in der /tmp.

  Die Arbeiten direkt auf dem Router sind jetzt abgeschlossen.
- Nun kopieren Sie die eben erzeugte Tastaturtabelle in Ihr fli4l-Verzeichnis unter opt/etc/<locale>.map. Wenn Sie jetzt `KEYBOARD_LOCALE='<locale>'` setzen wird beim nächsten Buildprozess Ihre neu erzeugtes Tastaturlayout benutzt.
- Vergessen Sie nicht `OPT_MAKEKBL` wieder auf 'no' zu setzen.

3.8. Ethernet/Token-Ring-Netzwerkkarten-Treiber

NET_DRV_N Anzahl der benötigten Netzwerkkarten-Treiber.

Wird der Router nur für ISDN verwendet, so gibt es in der Regel nur eine Netzwerkkarte, der Standard-Wert ist also 1. Bei DSL werden jedoch meistens zwei Netzwerkkarten verwendet.

Hierbei muss man zwei Fälle unterscheiden:

1. Beide Karten sind vom gleichen Typ. Dann muss nur ein Treiber geladen werden, der dann beide Karten anspricht, also `NET_DRV_N='1'`.
2. Es werden zwei verschiedene Karten verwendet, dann muss man hier '2' angeben und den Treiber für die zweite Karte angeben.

NET_DRV_x NET_DRV_x_OPTION Hier wird der Treiber für die Netzwerkkarte angegeben.[1]

Über die Variable `NET_DRV_1` wird standardmäßig der Treiber für eine NE2000-kompatible Netzwerkkarte geladen. Weitere verfügbare Treiber für Netzwerkkartenfamilien sind in den Tabellen 3.3 und 3.4 eingetragen.

[1] Für Token-Ring ist die Dokumentation etwas dünn, hier wäre ein wenig Dokumentation eines Token-Ring-Nutzers erforderlich. Im Anhang ist ein Auszug der Configure.help-Datei des Linux-Kerns zu finden, das als Starthilfe dienen kann ...

Die 3COM EtherLinkIII (3c509) muss über das Dostool 3c509cfg.exe (beziehbar von ftp://ftp.ihg.uni-duisburg.de/Hardware/3com/3C5x9n/3C5X9CFG.EXE)

konfiguriert werden. Dabei sollten IRQ und I/O-Port und gegebenenfalls auch der Anschluss (BNC/TP) eingestellt werden. Der Eintrag `NET_DRV_x_OPTION=`" kann in der Regel leer bleiben.

Bei manchen ISA-Karten braucht der Treiber zusätzliche Informationen, um die Karte zu finden, z.B. die I/O-Adresse. Bei NE2000-kompatiblen ISA-Karten und bei der EtherExpress16 ist dies zum Beispiel der Fall.

Hier ist z.B.

```
NET_DRV_x_OPTION='io=0x340'
```

zu setzen (oder der entsprechende numerische Wert).

Ist keine Option nötig, kann die Variable leer gelassen werden.

Sind mehrere Optionen nötig, sind diese durch Leerzeichen zu trennen, z.B.

```
NET_DRV_x_OPTION='irq=9 io=0x340'
```

Werden zwei identische Netzwerkkarten verwendet, z.B. NE2000-ISA-Karten, müssen die verschiedenen I/O-Werte durch Komma getrennt werden, also

```
NET_DRV_x_OPTION='io=0x240,0x300'
```

Die beiden IO-Werte müssen durch Komma ohne Blank getrennt werden!

Dieses funktioniert nicht bei allen Netzwerkkarten-Treibern. Einige muss man auch doppelt laden, also dann `NET_DRV_N='2'`. In diesem Fall müssen aber mit der Option "-o" verschiedene Namen vergeben werden, z.B.

```
NET_DRV_N='2'
NET_DRV_1='3c503'
NET_DRV_1_OPTION='-o 3c503-0 io=0x280'
NET_DRV_2='3c503'
NET_DRV_2_OPTION='-o 3c503-1 io=0x300'
```

Da ich in meinem fli4l-Router keine 2 Netzkwerkkarten habe, kann ich leider nicht mehr dazu sagen. Mein Tip: erst die Komma-Methode ausprobieren, danach das mehrfache Laden mit Option "-o" versuchen.

Nachfolgend noch einige Beispiele von Netzwerkkarten:

- 1 x NE2000 ISA

  ```
  NET_DRV_1='ne'
  NET_DRV_1_OPTION='io=0x340'
  ```

- 1 x 3COM EtherLinkIII (3c509)

  ```
  NET_DRV_1='3c509'
  NET_DRV_1_OPTION=''
  ```

Siehe zu dieser Karte auch:

http://extern.fli4l.de/fli4l_faqengine/faq.php?display=faq&faqnr=132&catnr=7&prog=1
http://extern.fli4l.de/fli4l_faqengine/faq.php?display=faq&faqnr=133&catnr=7&prog=1
http://extern.fli4l.de/fli4l_faqengine/faq.php?display=faq&faqnr=135&catnr=7&prog=1

- 2 x NE2000 ISA

```
NET_DRV_1='ne'
NET_DRV_1_OPTION='io=0x320,0x340'
```

  Oft muss man hier noch die IRQ-Werte mit angeben, also

```
NET_DRV_1_OPTION='io=0x320,0x340 irq=3,5'
```

  Man sollte es hier zunächst ohne Angabe der Interrupts probieren und lediglich dann die Interrupts eintragen, wenn ohne Angabe der Interrupts die Netzwerkkarten nicht gefunden werden.

- 2 x NE2000 PCI

```
NET_DRV_1='ne2k-pci'
NET_DRV_1_OPTION=''
```

- 1 x NE2000 ISA, 1 x NE2000 PCI

```
NET_DRV_1='ne'
NET_DRV_1_OPTION='io=0x340'
NET_DRV_2='ne2k-pci'
NET_DRV_2_OPTION=''
```

- 1 x SMC WD8013, 1 x NE2000 ISA

```
NET_DRV_1='wd'
NET_DRV_1_OPTION='io=0x270'
NET_DRV_2='ne2k'
NET_DRV_2_OPTION='io=0x240'
```

Vollständige Listen der möglichen Treiber finden Sie in der Tabelle der möglichen Kartentreiber und in der Tabelle der möglichen WLAN-Kartentreiber.

Falls Sie ein dummy-Device brauchen, verwenden Sie 'dummy' für `NET_DRV_x` *und* `IP_NET_x_DEV` *(Seite 39)='dummy<Nummer>' als Device-Name.*

Kernels	Bus	NET_DRV_x	Kartenfamilie
–	–	dummy	Dummy
2.6.32.59 2.6.16.62	devname	tun	Universal TUN/TAP device
2.6.32.59 2.6.16.62	eisa,isa	3c509	3Com Etherlink III (3c509, 3c509B, 3c529, 3c579) ethernet
2.6.32.59 2.6.16.62	eisa,pci	3c59x	3Com 3c59x/3c9xx ethernet
2.6.32.59 2.6.16.62	eisa,pci	hp100	HP CASCADE Architecture Driver for 100VG-AnyLan Network Adapters
2.6.32.59 2.6.16.62	isa	3c501	3Com Etherlink I
2.6.32.59 2.6.16.62	isa	3c503	3Com EtherLink II, II/16
2.6.32.59 2.6.16.62	isa	3c505	3COM Etherlink Plus
2.6.32.59 2.6.16.62	isa	3c507	3COM Etherlink 16

Kernels	Bus	NET_DRV_x	Kartenfamilie
2.6.32.59 2.6.16.62	isa	3c515	3Com 3c515 Corkscrew
2.6.32.59 2.6.16.62	isa	82596	i82596
2.6.32.59 2.6.16.62	isa	ac3200	Ansel AC3200 Eethernet
2.6.32.59 2.6.16.62	isa	at1700	AT1700 (Fujitsu 86965)
2.6.32.59 2.6.16.62	isa	cs89x0	cs89x0 based Cards
2.6.32.59 2.6.16.62	isa	e2100	Cabletron E2100 ethernet
2.6.32.59 2.6.16.62	isa	eepro	Intel i82595 EtherExpressPro10/10+
2.6.32.59 2.6.16.62	isa	eexpress	Intel EtherExpress16
2.6.32.59 2.6.16.62	isa,eisa	de4x5	Digital DE425, DE434, DE435, DE450, DE500
2.6.32.59 2.6.16.62	isa,eisa	depca	DEPCA, DE10x, DE200, DE201, DE202, DE422
2.6.32.59 2.6.16.62	isa	eth16i	ICL EtherTeam 16i/32
2.6.32.59 2.6.16.62	isa	ewrk3	EtherWORKS 3 ISA (DE203, DE204, DE205)
2.6.32.59 2.6.16.62	isa	hp	HP PC-LAN ethernet
2.6.32.59 2.6.16.62	isa	hp-plus	HP PC-LAN+ ethernet
2.6.32.59 2.6.16.62	isa,isa	ne	NE1000/NE2000 ISA/PnP Ethernet
2.6.32.59 2.6.16.62	isa,isa	smc-ultra	SMC Ultra/EtherEZ ISA/PnP Ethernet
2.6.32.59 2.6.16.62	isa	lance	AMD LANCE and PCnet (AT1500, NE2100)
2.6.32.59 2.6.16.62	isa	ni5010	MiCom-Interlan NI5010
2.6.32.59 2.6.16.62	isa	ni52	NI5210 card (i82586 Ethernet chip)
2.6.32.59 2.6.16.62	isa	ni65	AMD Lance Am7990
2.6.32.59	isa	proteon	Proteon 1392, 1932+ Token Ring
2.6.32.59	isa	seeq8005	Fujitsu MB86950, MB86960 or compatible controllers (Seeq 8005)
2.6.32.59	isa	skisa	SysKonnect TR4/16(+) (SK-4190)
2.6.32.59 2.6.16.62	isa	smc9194	SMC's 9000 series of Ethernet cards
2.6.32.59	isa	smctr	SMC TokenCard Elite (8115T) and SMC TokenCard Elite/A (8115T/A)
2.6.32.59 2.6.16.62	isa	wd	Western Digital wd8003/wd8013 ; SMC Elite, Elite16 ethernet
2.6.32.59	of	ethoc	OpenCores Ethernet MAC
2.6.32.59	pci	3c359	3Com 3C359 Velocity XL Token Ring Adapter
2.6.32.59	pci	8139cp	RealTek RTL-8139C+ series 10/100 PCI Ethernet
2.6.32.59 2.6.16.62	pci	8139too	RealTek RTL-8139 Fast Ethernet
2.6.32.59	pci	abyss	Madge Smart 16/4 PCI Mk2
2.6.32.59 2.6.16.62	pci	acenic	AceNIC/3C985/GA620 Gigabit Ethernet
2.6.32.59 2.6.16.62	pci	amd8111e	AMD8111 based 10/100 Ethernet Controller
2.6.32.59	pci	atl1	Atheros L1 Gigabit Ethernet
2.6.32.59	pci	atl1c	Atheros 1000M Ethernet Network
2.6.32.59	pci	atl1e	Atheros 1000M Ethernet Network
2.6.32.59	pci	atl2	Atheros Fast Ethernet Network
2.6.32.59 2.6.16.62	pci	b44	Broadcom 44xx/47xx 10/100 PCI ethernet
2.6.32.59	pci	be2net	ServerEngines BladeEngine 10Gbps NIC Driver 4.0.100u
2.6.32.59 2.6.16.62	pci	bnx2	Broadcom NetXtreme II BCM5706/5708/5709/5716
2.6.32.59	pci	bnx2x	Broadcom NetXtreme II BCM57710/57711/57711E/57712/57712_MF/57800/57800_MF/57810/57810_MF/57840/57840_MF
2.6.32.59 2.6.16.62	pci	cassini	Sun Cassini(+) ethernet
2.6.32.59 2.6.16.62	pci	cxgb	Chelsio 10Gb Ethernet
2.6.32.59	pci	cxgb3	Chelsio T3 Network
2.6.32.59 2.6.16.62	pci	de2104x	Intel/Digital 21040/1 series PCI Ethernet
2.6.32.59 2.6.16.62	pci	dl2k	D-Link DL2000-based Gigabit Ethernet Adapter
2.6.32.59 2.6.16.62	pci	dmfe	Davicom DM910X fast ethernet
2.6.32.59 2.6.16.62	pci	e100	Intel(R) PRO/100 Network
2.6.32.59 2.6.16.62	pci	e1000	Intel(R) PRO/1000 Network
2.6.32.59	pci	e1000e	Intel(R) PRO/1000 Network
2.6.16.62	pci	eepro100	Intel i82557/i82558/i82559 PCI EtherExpressPro

Kernels	Bus	NET_DRV_x	Kartenfamilie
2.6.32.59	pci	ems_pci	Socket-CAN driver for EMS CPC-PCI/PCIe/104P CAN cards
2.6.32.59	pci	enic	Cisco VIC Ethernet NIC
2.6.32.59 2.6.16.62	pci	epic100	SMC 83c170 EPIC series Ethernet
2.6.32.59 2.6.16.62	pci	fealnx	Myson MTD-8xx 100/10M Ethernet PCI Adapter
2.6.32.59 2.6.16.62	pci	forcedeth	Reverse Engineered nForce ethernet
2.6.32.59 2.6.16.62	pci	hamachi	Packet Engines 'Hamachi' GNIC-II Gigabit Ethernet
2.6.32.59	pci	igb	Intel(R) Gigabit Ethernet Network
2.6.32.59	pci	igbvf	Intel(R) 82576 Virtual Function Network
2.6.32.59	pci	ipg	IC Plus IP1000 Gigabit Ethernet Adapter Linux
2.6.32.59 2.6.16.62	pci	ixgb	Intel(R) PRO/10GbE Network
2.6.32.59	pci	ixgbe	Intel(R) 10 Gigabit PCI Express Network
2.6.32.59	pci	jme	JMicron JMC2x0 PCI Express Ethernet
2.6.32.59	pci	kvaser_pci	Socket-CAN driver for KVASER PCAN PCI cards
2.6.32.59	pci	lanstreamer	IBM PCI tokenring cards based on the LanStreamer MPC
2.6.32.59	pci	mlx4_core	Mellanox ConnectX HCA low-level
2.6.32.59	pci	myri10ge	Myricom 10G driver (10GbE)
2.6.32.59 2.6.16.62	pci	natsemi	National Semiconductor DP8381x series PCI Ethernet
2.6.32.59 2.6.16.62	pci	ne2k-pci	PCI NE2000 clone
2.6.32.59	pci	netxen_nic	QLogic/NetXen (1/10) GbE Intelligent Ethernet
2.6.32.59	pci	niu	Sun Neptun Ethernet
2.6.32.59 2.6.16.62	pci	ns83820	National Semiconductor DP83820 10/100/1000
2.6.32.59	pci	olympic	Olympic PCI/Cardbus Chipset
2.6.32.59 2.6.16.62	pci	pcnet32	PCnet32 and PCnetPCI based ethercards
2.6.32.59	pci	qla3xxx	QLogic ISP3XXX Network Driver v2.03.00-k5
2.6.32.59	pci	qlge	QLogic 10 Gigabit PCI-E Ethernet
2.6.32.59	pci	r6040	RDC R6040 NAPI PCI FastEthernet
2.6.32.59 2.6.16.62	pci	r8169	RealTek RTL-8169 Gigabit Ethernet
2.6.32.59 2.6.16.62	pci	s2io	Neterion 10GbE Server NIC
2.6.32.59	pci	sc92031	Silan SC92031 PCI Fast Ethernet Adapter
2.6.32.59	pci	sfc	Solarflare Communications network
2.6.32.59 2.6.16.62	pci	sis190	SiS sis190/191 Gigabit Ethernet
2.6.32.59 2.6.16.62	pci	sis900	SiS 900 PCI Fast Ethernet
2.6.16.62	pci	sk98lin	SysKonnect SK-NET Gigabit Ethernet SK-98xx
2.6.32.59 2.6.16.62	pci	skge	SysKonnect Gigabit Ethernet
2.6.32.59 2.6.16.62	pci	sky2	Marvell Yukon 2 Gigabit Ethernet
2.6.32.59	pci	smsc9420	SMSC LAN9420
2.6.32.59 2.6.16.62	pci	sundance	Sundance Alta Ethernet
2.6.32.59 2.6.16.62	pci	sungem	Sun GEM Gbit ethernet
2.6.32.59 2.6.16.62	pci	sunhme	Sun HappyMealEthernet(HME) 10/100baseT ethernet
2.6.32.59	pci	tehuti	Tehuti Networks(R) Network
2.6.32.59 2.6.16.62	pci	tg3	Broadcom Tigon3 ethernet
2.6.32.59 2.6.16.62	pci	tlan	TI ThunderLAN based ethernet PCI adapters
2.6.32.59	pci	tmspci	TMS380-based token ring cards
2.6.32.59 2.6.16.62	pci	tulip	Digital 21*4* Tulip ethernet
2.6.32.59 2.6.16.62	pci	typhoon	3Com Typhoon Family (3C990, 3CR990, and variants)
2.6.32.59 2.6.16.62	pci	uli526x	ULi M5261/M5263 fast ethernet
2.6.32.59 2.6.16.62	pci	via-rhine	VIA Rhine PCI Fast Ethernet
2.6.32.59 2.6.16.62	pci	via-velocity	VIA Networking Velocity Family Gigabit Ethernet Adapter
2.6.32.59	pci	vmxnet3	VMware vmxnet3 virtual NIC
2.6.32.59	pci	vxge	Neterion's X3100 Series 10GbE PCIe I/OVirtualized Server Adapter
2.6.32.59 2.6.16.62	pci	winbond-840	Winbond W89c840 Ethernet
2.6.32.59 2.6.16.62	pci	xircom_cb	Xircom Cardbus ethernet
2.6.16.62	pci	xircom_tulip_cb	Xircom CBE-100 ethernet

Kernels	Bus	NET_DRV_x	Kartenfamilie
2.6.32.59 2.6.16.62	pci	yellowfin	Packet Engines Yellowfin G-NIC Gigabit Ethernet
2.6.32.59 2.6.16.62	pcmcia	3c574_cs	3Com 3c574 series PCMCIA ethernet
2.6.32.59 2.6.16.62	pcmcia	3c589_cs	3Com 3c589 series PCMCIA ethernet
2.6.32.59 2.6.16.62	pcmcia	axnet_cs	Asix AX88190 PCMCIA ethernet
2.6.32.59 2.6.16.62	pcmcia	fmvj18x_cs	fmvj18x and compatible PCMCIA ethernet
2.6.32.59	pcmcia	ibmtr_cs	IBM PCMCIA Token-Ring Card
2.6.32.59 2.6.16.62	pcmcia	nmclan_cs	New Media PCMCIA ethernet
2.6.32.59 2.6.16.62	pcmcia	pcnet_cs	NE2000 compatible PCMCIA ethernet
2.6.32.59 2.6.16.62	pcmcia	smc91c92_cs	SMC 91c92 series PCMCIA ethernet
2.6.32.59 2.6.16.62	pcmcia	xirc2ps_cs	Xircom PCMCIA ethernet
2.6.32.59	usb	asix	ASIX AX8817X based USB 2.0 Ethernet Devices
2.6.32.59	usb	catc	CATC EL1210A NetMate USB Ethernet
2.6.32.59	usb	cdc_eem	USB CDC EEM
2.6.32.59	usb	cdc_ether	USB CDC Ethernet devices
2.6.32.59	usb	cdc-phonet	USB CDC Phonet host interface
2.6.32.59	usb	cdc_subset	Simple 'CDC Subset' USB networking links
2.6.32.59	usb	dm9601	Davicom DM9601 USB 1.1 ethernet devices
2.6.32.59	usb	gl620a	GL620-USB-A Host-to-Host Link cables
2.6.32.59	usb	hso	USB High Speed Option
2.6.32.59	usb	int51x1	Intellon usb powerline adapter
2.6.32.59	usb	kaweth	KL5USB101 USB Ethernet
2.6.32.59	usb	mcs7830	USB to network adapter MCS7830)
2.6.32.59	usb	net1080	NetChip 1080 based USB Host-to-Host Links
2.6.32.59	usb	pegasus	Pegasus/Pegasus II USB Ethernet
2.6.32.59	usb	plusb	Prolific PL-2301/2302/25A1 USB Host to Host Link
2.6.32.59	usb	rndis_host	USB Host side RNDIS
2.6.32.59	usb	rtl8150	rtl8150 based usb-ethernet
2.6.32.59	usb	smsc95xx	SMSC95XX USB 2.0 Ethernet Devices
2.6.32.59	usb	zaurus	Sharp Zaurus PDA, and compatible products
2.6.32.59	virtio	virtio_net	Virtio network

Tabelle 3.3.: Tabelle der möglichen Netzwerkkartentreiber

Kernels	Bus	NET_DRV_x	Kartenfamilie
2.6.32.59 2.6.16.62	isa,pci	airo	Cisco/Aironet 802.11 wireless ethernet cards
2.6.32.59	isa	wavelan	AT&T GIS WaveLAN card
2.6.32.59	pci	adm8211	IEEE 802.11b wireless cards based on ADMtek ADM8211
2.6.32.59	pci	ath5k	5xxx series of Atheros 802.11 wireless LAN cards
2.6.32.59	pci	ath9k	Atheros 802.11n wireless LAN cards
2.6.32.59 2.6.16.62	pci	ath_pci	Atheros 802.11 wireless LAN cards
2.6.32.59 2.6.16.62	pci	atmel_pci	Atmel at76c50x 802.11 wireless ethernet cards
2.6.32.59	pci	b43legacy	Broadcom B43legacy wireless
2.6.32.59 2.6.16.62	pci	hostap_pci	Intersil Prism2.5-based 802.11 wireless LAN PCI cards
2.6.32.59 2.6.16.62	pci	hostap_plx	Intersil Prism2-based 802.11 wireless LAN cards (PLX)
2.6.32.59 2.6.16.62	pci	ipw2100	Intel(R) PRO/Wireless 2100 Network
2.6.32.59 2.6.16.62	pci	ipw2200	Intel(R) PRO/Wireless 2200/2915 Network
2.6.32.59	pci	iwl3945	Intel(R) PRO/Wireless 3945ABG/BG Network Connection driver for Linux
2.6.32.59	pci	iwlagn	Intel(R) Wireless WiFi Link AGN driver for Linux
2.6.32.59	pci	mwl8k	Marvell TOPDOG(R) 802.11 Wireless Network

Kernels	Bus	NET_DRV_x	Kartenfamilie
2.6.32.59 2.6.16.62	pci	orinoco_nortel	wireless LAN cards using the Nortel PCI bridge
2.6.32.59	pci	orinoco_pci	wireless LAN cards using direct PCI interface
2.6.32.59 2.6.16.62	pci	orinoco_plx	wireless LAN cards using the PLX9052 PCI bridge
2.6.32.59 2.6.16.62	pci	orinoco_tmd	wireless LAN cards using the TMD7160 PCI bridge
2.6.32.59	pci	p54pci	Prism54 PCI wireless
2.6.32.59	pci,pcmcia	b43	Broadcom B43 wireless
2.6.32.59 2.6.16.62	pci	prism54	The Prism54 802.11 Wireless LAN adapter
2.6.32.59	pci	rt2400pci	Ralink RT2400 PCI & PCMCIA Wireless LAN
2.6.32.59	pci	rt2500pci	Ralink RT2500 PCI & PCMCIA Wireless LAN
2.6.32.59	pci	rt61pci	Ralink RT61 PCI & PCMCIA Wireless LAN
2.6.32.59	pci	rtl8180	RTL8180 / RTL8185 PCI wireless
2.6.32.59 2.6.16.62	pcmcia	airo_cs	Cisco/Aironet 802.11 wireless ethernet cards
2.6.32.59 2.6.16.62	pcmcia	atmel_cs	Atmel at76c50x 802.11 wireless ethernet cards
2.6.32.59 2.6.16.62	pcmcia	hostap_cs	Intersil Prism2-based 802.11 wireless LAN cards (PC Card)
2.6.32.59	pcmcia	libertas_cs	Marvell 83xx compact flash WLAN cards
2.6.32.59	pcmcia	netwave_cs	Netwave AirSurfer Wireless LAN PC Card
2.6.32.59 2.6.16.62	pcmcia	orinoco_cs	PCMCIA Lucent Orinoco, Prism II based and similar wireless cards
2.6.32.59 2.6.16.62	pcmcia	ray_cs	Raylink/WebGear wireless LAN
2.6.32.59	pcmcia	wavelan_cs	AT&T GIS WaveLAN pcmcia card
2.6.32.59 2.6.16.62	pcmcia	wl3501_cs	Planet wl3501 wireless
2.6.32.59	usb	ar9170usb	Atheros AR9170 802.11n USB wireless
2.6.32.59	usb	at76c50x-usb	Atmel at76x USB Wireless LAN
2.6.32.59	usb	i2400m-usb	USB based Intel Wireless WiMAX Connection 2400M (5x50 & 6050)
2.6.32.59	usb	p54usb	Prism54 USB wireless
2.6.32.59	usb	rndis_wlan	RNDIS based USB Wireless adapters
2.6.32.59	usb	rt2500usb	Ralink RT2500 USB Wireless LAN
2.6.16.62	usb	rt2570	Ralink RT2570 usb 802.11g WLAN driver 1.0.0 - CVS 2009041204
2.6.32.59	usb	rt2800usb	Ralink RT2800 USB Wireless LAN
2.6.32.59	usb	rt73usb	Ralink RT73 USB Wireless LAN
2.6.32.59	usb	rtl8187	RTL8187/RTL8187B USB wireless
2.6.32.59	usb	usb8xxx	8388 USB WLAN
2.6.32.59	usb	zd1201	ZyDAS ZD1201 based USB Wireless adapters
2.6.32.59	usb	zd1211rw	USB driver for devices with the ZD1211 chip
2.6.32.59	virtual	mac80211_hwsim	Software simulator of 802.11 radio(s) for mac80211

Tabelle 3.4.: Tabelle der möglichen WLAN-Kartentreiber

3.9. Netzwerke

IP_NET_N Anzahl der Netzwerke, die an das IP-Protokoll gebunden werden sollen. Im Normalfall also '1'. Falls es keine Netzwerke geben sollte oder sie über einen anderen Weg konfiguriert werden, und daher `IP_NET_N` auf 0 gesetzt wird, wird beim Bauen der Archive eine Warnung ausgegeben. Diese kann man durch setzen von `IGNOREIPNETWARNING`='yes' aussschalten.

IP_NET_x Standardwert: `IP_NET_1='192.168.6.1/24'`

Die IP-Adresse und die Netzmaske in CIDR[2]-Schreibweise des x'ten Devices im fli4l-Router. Wird die IP-Adresse dynamisch durch einen DHCP-Klienten zugewiesen, kann hier auch 'dhcp' als Wert eingetragen werden.

Nachfolgende Tabelle zeigt CIDR und Punkt-Schreibweise (DOT Notation).

CIDR	Netzmaske	Anzahl IPs
/8	255.0.0.0	16777216
/16	255.255.0.0	65536
/23	255.255.254.0	512
/24	255.255.255.0	256
/25	255.255.255.128	128
/26	255.255.255.192	64
/27	255.255.255.224	32
/28	255.255.255.240	16
/29	255.255.255.248	8
/30	255.255.255.252	4
/31	255.255.255.254	2
/32	255.255.255.255	1

Anmerkung: Da jeweils eine IP für die Netzwerk- und Broadcast-Adresse reserviert sind, errechnet sich die maximale Anzahl der Hosts im Netzwerk zu: `Anzahl_Hosts = Anzahl_IPs - 2`. Die kleinste mögliche Netzmaske wäre also `/30`, entspricht 4 IPs und somit 2 möglichen Hosts.

IP_NET_x_DEV Erforderlich: Device-Name der Netzwerkkarte.

Ab Version 2.1.8 ist die Angabe des verwendeten Device zwingend erforderlich! Die Namen der Devices beginnen in den meisten Fällen mit `'eth'` gefolgt von einer Zahl. Die erste vom System erkannte Netzwerkkarte bekommt den Namen `'eth0'`, die zweite `'eth1'` usw.
Bei Token-Ring Netzwerkken lautet die Bezeichung `'tr0'`, `'tr1'` usw.

Beispiel:

```
IP_NET_1_DEV='eth0'
IP_NET_1_DEV='tr0'
```

fli4l beherrscht auch IP-Aliasing, also die Zuweisung von mehreren IPs auf eine Netzwerkkarte. Zusätzliche IPs definiert man einfach mit einem weiteren Netzwerk auf dem selben Device. Beim Überprüfen der Konfiguration informiert mkfli4l darüber, dass man ein solches Alias definiert — diese Warnung kann man dann ignorieren.

Beispiel:

```
IP_NET_1='192.168.6.1/24'
IP_NET_1_DEV='eth0'
IP_NET_2='192.168.7.1/24'
IP_NET_2_DEV='eth0'
```

[2] Classless Inter-Domain Routing

Standard-Einstellung: `IP_NET_1_DEV='eth0'`

IP_NET_x_MAC Optional: MAC-Adresse der Netzwerkkarte.

Hier kann die Hardwareadresse (MAC) der Netzwerkkarte angepasst werden. Dies ist zum Beispiel nützlich, wenn Sie einen DHCP-Provider nutzen wollen, der eine bestimmte MAC-Adresse erwartet. Wird `IP_NET_x_MAC` leer gelassen oder garnicht angegeben, so wird die voreingestellte MAC-Adresse der Netzwerkkarte verwendet. Die allermeisten Nutzer werden diese Variable nicht benötigen.

Beispiel:

```
IP_NET_1_MAC='01:81:42:C2:C3:10'
```

Standard-Einstellung: `IP_NET_1_MAC=''`

IP_NET_x_NAME Optional: Der IP-Adresse der Netzwerkkarte einen Namen geben.

Bei umgekehrter Namensauflösung der IP-Adresse der Netzwerkkarte erscheint standardmässig ein Name der Form 'fli4l-ethx.<domain>'. In `IP_NET_x_NAME` kann man selbst einen anderen Namen angeben. Dieser Name wird dann bei umgekehrter Namensauflösung angezeigt. Bei einer öffentlichen IP-Adresse kann man so erreichen, dass immer der öffentliche Name aufgelöst wird.

Beispiel:

```
IP_NET_2='80.126.238.229/32'
IP_NET_2_NAME='ajv.xs4all.nl'
```

Standard-Einstellung: `IP_NET_x_NAME=''`

IP_NET_x_TYPE Siehe DMZ – Demilitarisierte Zone. (Seite 71)

IP_NET_x_COMMENT Optional: Die Angabe dient dazu, einem Device einen 'aussagekräftigen' Namen zu geben. Dieser kann dann in Paketen wie z.B. rrdtool zur Identifikation des Netzwerks verwendet werden.

Standard-Einstellung: `IP_NET_x_COMMENT=''`

3.10. Zusätzliche Routen (optional)

IP_ROUTE_N Anzahl von zusätzlichen Netzwerkrouten. Zusätzliche Netzwerkrouten sind zum Beispiel dann erforderlich, wenn sich im LAN weitere Router befinden, über die andere Netzwerke erreichbar sein sollen.

Im Normalfall ist die Angabe von weiteren Netzwerkrouten nicht erforderlich, daher ist die

Standard-Einstellung: `IP_ROUTE_N='0'`

IP_ROUTE_x Die zusätzlichen Routen `IP_ROUTE_1`, `IP_ROUTE_2`, ... haben folgenden Aufbau:

```
network/netmaskbits gateway
```

Dabei ist `network` die Netzwerkadresse, `/netmaskbits` die Netzmaske in CIDR (Seite 39) Schreibweise und `gateway` die Adresse des Rechners, der die Verbindung zu dem Netzwerk herstellt. Der Gateway-Rechner muss natürlich im gleichen Netzwerk, wie der fli4l-Router liegen! Ist z.B. das Netzwerk 192.168.7.0 mit der Netzwerkmaske 255.255.255.0 über das Gateway 192.168.6.99 erreichbar, dann lautet der Eintrag:

```
IP_ROUTE_N='1'
IP_ROUTE_1='192.168.7.0/24 192.168.6.99'
```

Wenn der fli4l-Router nicht als Internet-Router eingesetzt wird sondern nur als reiner Ethernetrouter kann man in einem IP_ROUTE_x Eintrag eine Default-Route angeben. Dazu wird analog zu der network/netmaskbits Schreibweise 0.0.0.0/0 eingetragen, so wie im Beispiel zu sehen ist.

```
IP_ROUTE_N='3'
IP_ROUTE_1='192.168.1.0/24 192.168.6.1'
IP_ROUTE_2='10.73.0.0/16 192.168.6.1'
IP_ROUTE_3='0.0.0.0/0 192.168.6.99'
```

3.11. Konfiguration des Paketfilters

Der von Fli4l verwendete Linux-Kern stellt einen Paketfilter zur Verfügung. Mit Hilfe dieses Paketfilters wird gesteuert, wer mit dem Router bzw. über ihn hinweg kommunizieren darf. Weiterhin können Dinge wie Port-Forwarding (ein an den Router gerichtetes Paket wird an einen anderen internen Rechner weitergereicht) und Masquerading (Pakete, die von einem Rechner hinter dem Router kommen, werden so verändert, dass sie so aussehen, als kämen sie vom Router selbst) realisiert werden. Für die Konfiguration gibt es zwei Varianten, die alte auch in 2.0.x vorhandene Variante und eine neue, die eine flexiblere Konfiguration des Paketfilters gestattet. Welche Variante genutzt wird, wird durch die Variable `PF_NEW_CONFIG` bestimmt. Standardmäßig steht `PF_NEW_CONFIG` auf `'yes'` und wählt damit die neue Konfiguration aus. Hier beschreiben wir allerdings zuerst die originale Konfiguration, die einfacher zu verstehen ist und im Anschluss daran die neue Konfiguration, die wesentlich flexibler ist, aber auch mehr Einarbeitung erfordert. Wählt man die originale Konfiguration aus, zeigt der Router beim Booten an, wie diese Konfiguration im neuen Format aussehen würde. Man kann sich das in aller Ruhe ansehen und evtl, bei Bedarf auf die neue, flexiblere Konfiguration wechseln.

Die Struktur des Paketfilters ist in Abbildung 3.1 angedeutet. Pakete kommen auf einem Netzwerkinterface rein und durchlaufen die Pre-Routing-Chain. Hier werden evtl. an den Router gerichtete Pakete an einen anderen Rechner weitergereicht, indem die Ziel-Ip-Adresse und der Zielport manipuliert werden. Ist das Paket an den Router gerichtet, wird es an die Input-Chain, andernfalls an die Forward-Chain weitergereicht. Beide Chains prüfen, ob das Paket zulässig ist. Wird das Paket akzeptiert, wird es an den lokalen Zielprozess zugestellt oder über die Post-Routing-Chain (in der das Masquerading stattfindet) an das Netzwerkinterface weitergereicht, über das das Paket sein Ziel erreichen kann.

3.12. Die alte Konfiguration des Paketfilters

Bei der Konfiguration des Paketfilters werden zwei Dinge unterschieden: Die kommunikation mit dem Router und die Kommunikation mit anderen Netzen über den Router hinweg. An

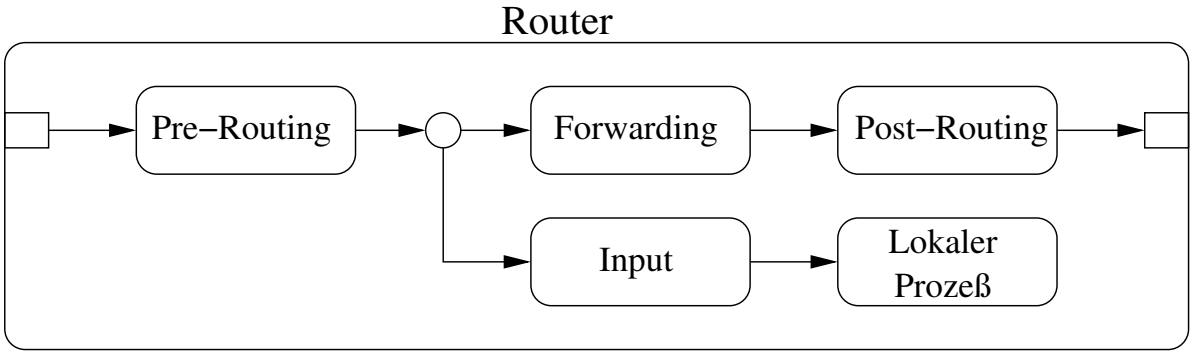


Abbildung 3.1.: Struktur des Paketfilters

allen Stellen, an denen Netzwerke, IP-Adressen oder Hosts angegeben werden müssen, kann man sich auch auf `IP_NET_x`, `IP_NET_x_IPADDR` oder via `'@hostname'` auf einen Host aus `HOST_x` beziehen.

3.12.1. Kommunikation über den Router hinweg

Der Paketfilter wird so konfiguriert, dass er generell nur Pakete von IP-Adressen akzeptiert, die ihm bekannt sind. Bekannt sind ihm IP-Adressen aus Subnetzen, die in folgenden Variablen angegeben wurden:

- `TRUSTED_NETS` – Subnetze, zwischen denen geforwarded wird und zwischen denen eine ungehinderte und unmaskierte Kommunikation möglich ist.
- `ROUTE_NETWORK` – Subnetze, die geforwardet werden; Pakete, die in diese Netze hineingehen, werden nicht maskiert.
- `MASQ_NETWORK` – Subnetze, die geforwarded und maskiert werden.

Weiterhin kennt der Router auch alle Rechner, zu denen Verbindungen aufgebaut wurden.

Zusätzlich wendet Fli4l auf die ihm bekannten IP-Adressen noch eine Black-/Whitelist (`FORWARD_HOST_x`) an und filtert bestimmte Ports heraus (`FORWARD_DENY_PORT_x`. Die Anwendung der Filter ist in Abbildung 3.2 dargestellt. Eine zutreffende Regel verzweigt im Bild nach oben bzw. nach unten, trifft die Regel nicht zu, wird die nächste genommen. Das wird durch den Pfeil nach rechts symbolisiert.

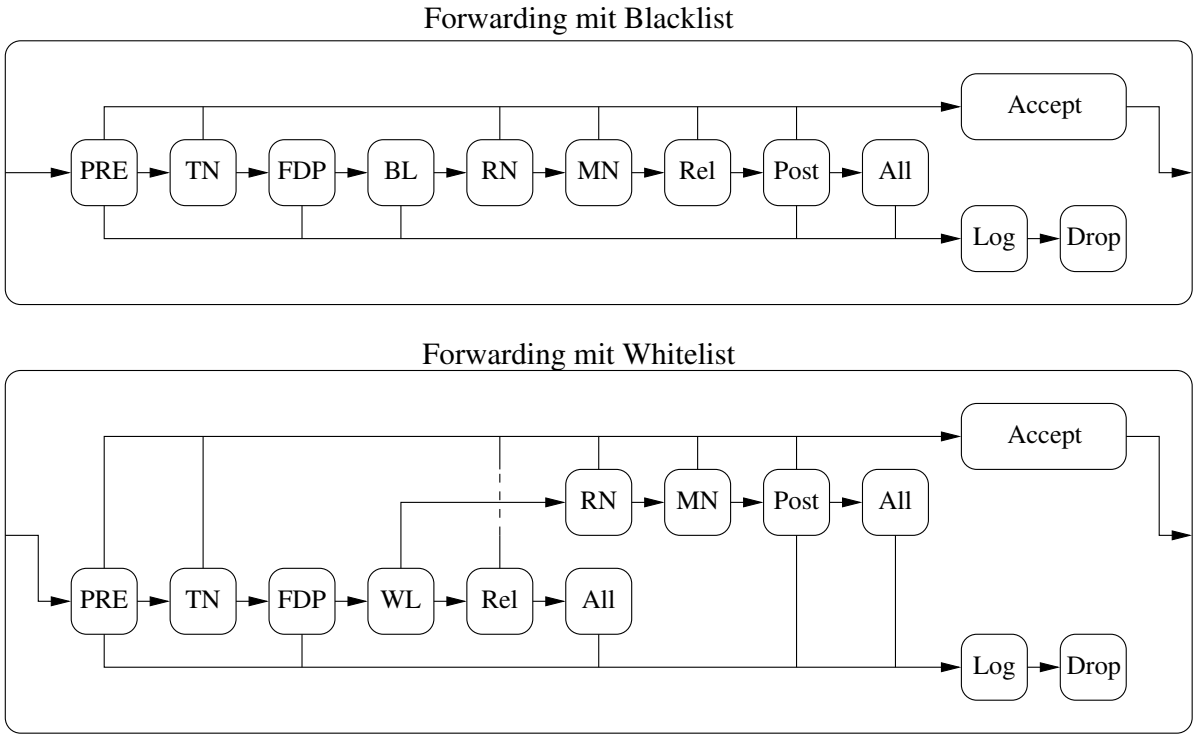
Die Konfigurationsvariablen werden im Folgenden genauer beschrieben.

MASQ_NETWORK Hier sind die Netzwerke anzugeben, die nach aussen hin maskiert werden sollen. Verwendet man nicht-offizielle IP-Adressen, wie z.B. 192.168.x.x, und soll der Router als Zugang ins Internet verwendet werden, müssen diese hier unbedingt angegeben werden.

Die Form ist:

```
NETZWERKNUMMER/ANZAHL-DER-GESETZTEN-BITS-IN-NETMASK
```

also z.B. für Netze der Form 192.168.x.0:



PRE/Post Regeln die vom Nutzer selbst geschrieben wurden und unter opt/etc/rc.d/fwrules.pre,post.* abgelegt wurden. Diese können die Pakete sowohl akzeptieren als auch verwerfen.

TN Kommunikation zwischen `TRUSTED_NETS` wird ohne weitere Filterung akzeptiert

FDP Pakete, die an in `FORWARD_DENY_PORT_x` spezifizierte Ports gerichtet werden, werden verworfen.

BL Kommt das Paket von einem in der Blacklist angegebenen Rechner, wird es verworfen.

WL Kommt das Paket von einem in der Whitelist angegeben Rechner, wird es erst einmal weiterbetrachtet (es muss zusätzlich noch aus einem der bekannten Netze stammen).

RN Kommt das Paket von einem Rechner aus einem in `ROUTE_NETWORK` angegebenen Subnetz, wird es akzeptiert.

MN Kommt das Paket von einem Rechner aus einem in `MASQ_NETWORK` angegebenen Subnetz, wird es akzeptiert.

Rel Gehört das Paket zu einer bereits aufgebauten Verbindung, wird es akzeptiert (egal, woher es kommt).

All Eine Regel, die auf alle Pakete zutrifft und dafür sorgt, dass das Paket verworfen wird.

Log Es wird für das Paket ein Eintrag im Syslog bzw. auf der Konsole erzeugt.

Drop Das Paket wird verworfen.

Accept Das Paket wird akzeptiert und damit weitergeleitet.

Abbildung 3.2.: Konfigurationsvariablen und ihre Anwendung

```
MASQ_NETWORK='192.168.6.0/24'
```

Hier sind also die ersten 24 Bits in der Netzwerkmaske gesetzt.

Sollen mehrere Netze maskiert werden, sind diese durch Leerzeichen zu trennen. Soll kein Netz maskiert werden, muss der Variablen-Inhalt leer sein.

Wie ein Paket aus einem in `MASQ_NETWORK` angegebenen Subnetz durch die Firewall läuft, ist in Abbildung 3.3 dargestellt.

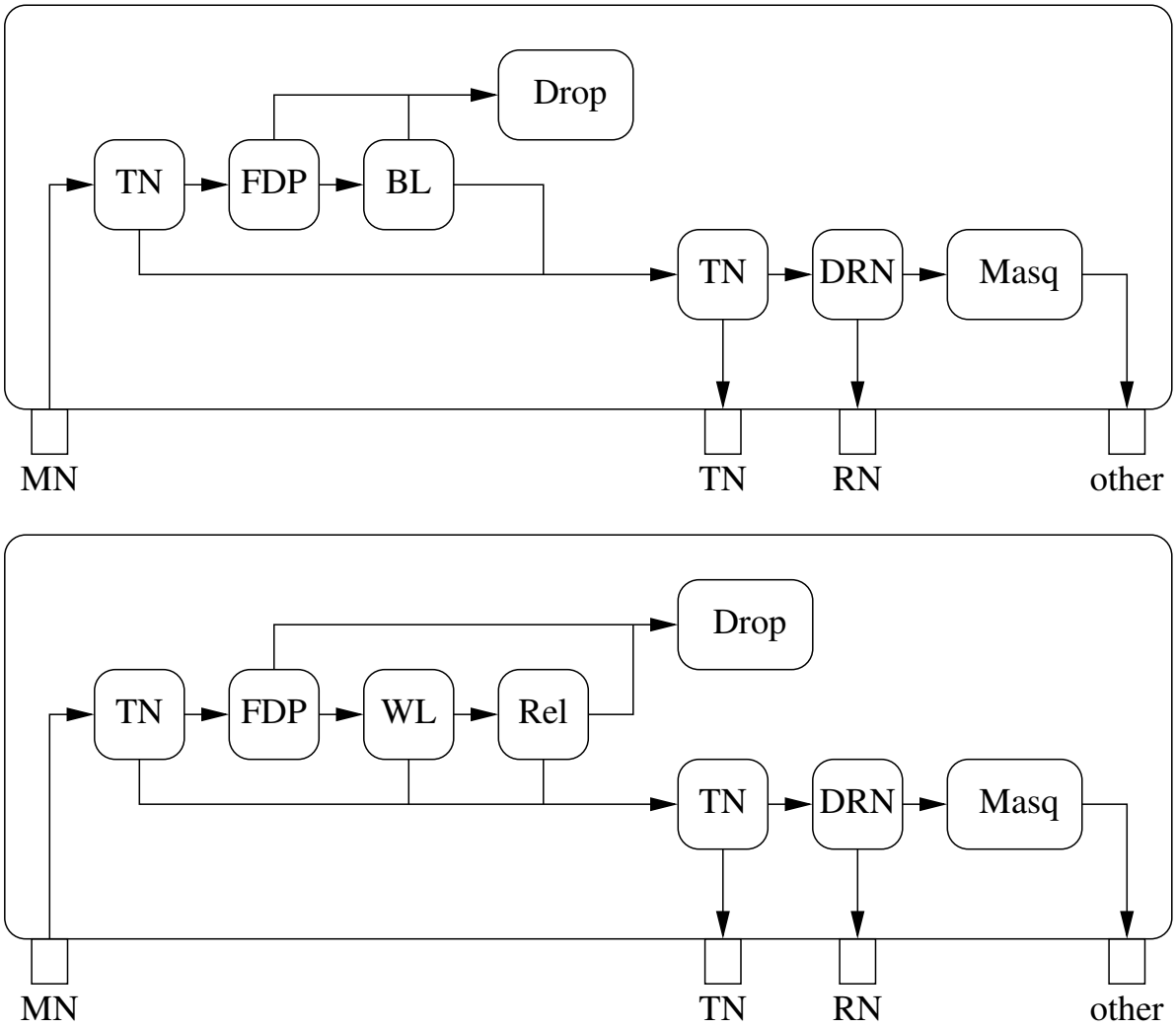


Abbildung 3.3.: Der Weg von Paketen aus einem MASQ_NETWORK durch den Paketfilter

ROUTE_NETWORK Pakete, die zu von Hosts in diesen Subnetzen aufgebauten Verbindungen gehören, werden vom Router geforwarded. Zusätzlich werden Pakete, die in diese Netze hineingehen, nicht maskiert.

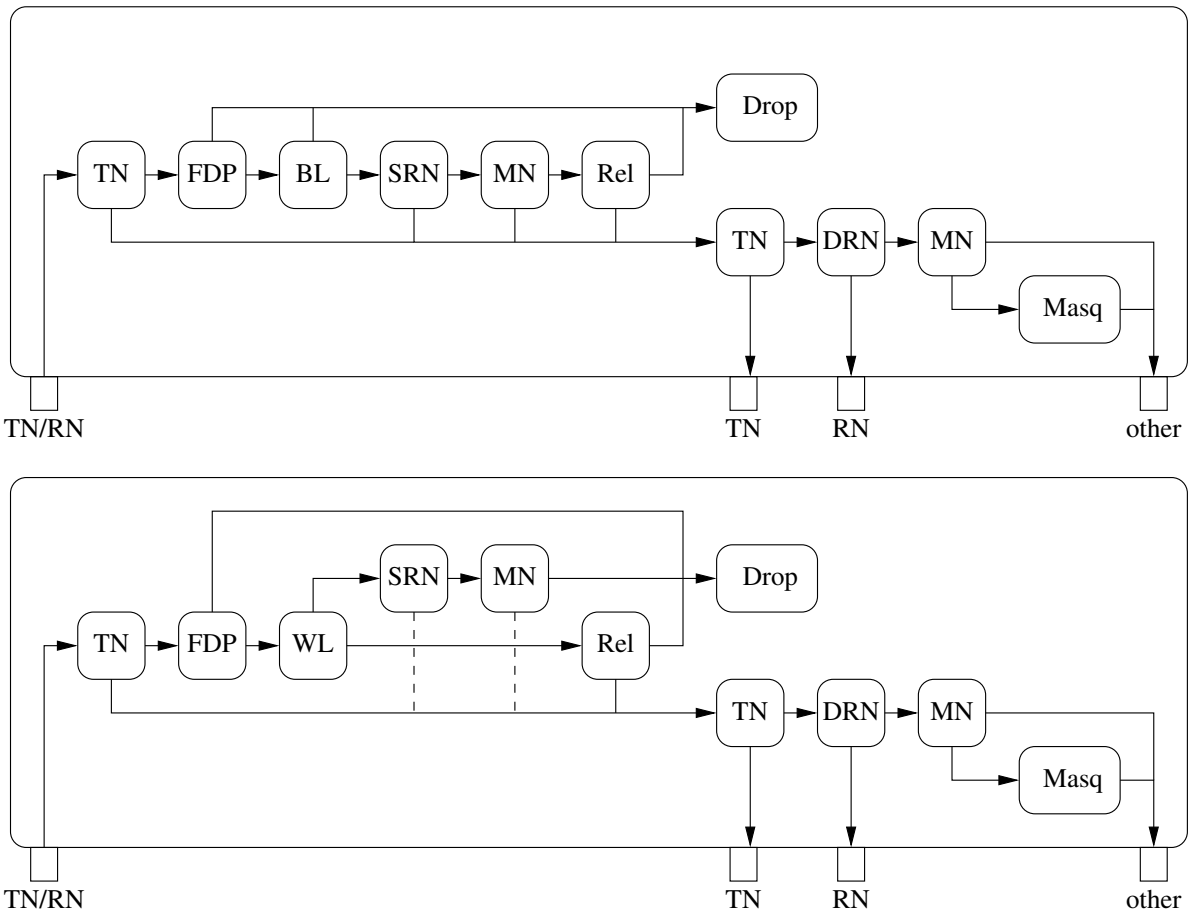
Dieses kann z.B. für eine Verbindung zur Firma sinnvoll sein, wo Mitarbeiter dieser Firma von dort Zugang in das eigene Netz haben sollen. Das Format ist dasselbe wie bei `MASQ_NETWORK`, also z.B.

```
ROUTE_NETWORK='192.168.1.0/24'
```

Auch können hier wieder mehrere Netze angegeben werden.

`ROUTE_NETWORK` kann in der Regel einfach leer bleiben.

Der Weg, den Pakete eines `ROUTE_NETWORK` nehmen, ist in Abbildung 3.4 dargestellt (es ist der gleiche wie ein Paket aus einem der `TRUSTED_NETS`).



SRN Das Paket stammt aus einem in `ROUTE_NETWORK` angegebenen Subnetz.

DRN Das Paket soll in ein in `ROUTE_NETWORK` angegebenes Subnetz geleitet werden.

Abbildung 3.4.: Der Weg von Paketen aus einem der TRUSTED_NETS bzw. ROUTE_NETWORK durch den Paketfilter

FORWARD_HOST_WHITE Es gibt Situationen, da möchte man entweder einigen Rechnern gezielt den Zugriff auf andere Netze erlauben (Whitelist) oder genau andersherum einigen Rechnern den Zugriff auf andere Netze verbieten (Blacklist). Der Paketfilter lässt dann lediglich Pakete der angegebenen Rechner durch bzw. blockiert genau diese. Setzt man `FORWARD_HOST_WHITE` auf yes, stellt die nachfolgende Aufzählung eine Whitelist dar, andernfalls eine Blacklist.

Will man also keinem Rechner die Kommunikation über den Router verbieten, definiert man eine leere Blacklist (FORWARD_HOST_WHITE='no' und `FORWARD_HOST_N`='0'). Dies ist auch die Standardkonfiguration.

FORWARD_HOST_N FORWARD_HOST_x Hier werden die Rechner der White- bzw. Blacklist aufgeführt. Z.B.

```
FORWARD_HOST_N='2'
FORWARD_HOST_1='192.168.6.5'
FORWARD_HOST_2='192.168.6.6'
```

Pakete dieser beiden Rechnern werden im Falle einer Whitelist als einzige vom Router weitergeleitet, im anderen Falle als einzige blockiert.

FORWARD_DENY_PORT_N FORWARD_DENY_PORT_x Hiermit kann man das Routing über bestimmte IP-Ports verhindern. Sinnvoll ist z.B. das Verbieten des Routings der Netbios-Ports 137-139. Damit wird nicht nur das Routing von IP-Paketen mit den angegebenen Ports nach "außen" unterbunden, sondern auch das Routing dieser Ports zwischen zwei LANs.

Betreibt man mehrere Netzwerkkarten für mehrere Subnetze und möchte, dass einige Clients aus einem Subnetz auf die unter Windows freigegebenen Verzeichnisse eines Clients aus einem anderen Subnetz zugreifen können, sollte das Forwarding der Netbios-Ports aber hier nicht unterbunden werden. In diesem Fall kann man mit `TRUSTED_NETS` (s.o.) Netze angeben, zwischen denen das Routing dieser Ports dennoch explizit erlaubt ist.

Ein Beispiel: Abschalten Zugriff auf Napster-Dienste:

```
FORWARD_DENY_PORT_N='7'                 # Anzahl
FORWARD_DENY_PORT_1='135-139 REJECT'    # NetBios-Ports 135 bis 139
FORWARD_DENY_PORT_2='445     REJECT'    # MS-DS
FORWARD_DENY_PORT_3='8888    REJECT'    # Napster Port 8888
FORWARD_DENY_PORT_4='7777    REJECT'    # Napster Port 7777
FORWARD_DENY_PORT_5='7000    REJECT'    # Napster Port 7000
FORWARD_DENY_PORT_6='4444    REJECT'    # Napster Port 4444
FORWARD_DENY_PORT_7='4200    REJECT'    # Napster Port 4200
```

TRUSTED_NETS Das Sperren des Routings der mittels `FORWARD_DENY_PORT_x` angegebenen Ports und die Black/Whitelist (`FORWARD_DENY_HOST`) kann mit Hilfe von `TRUSTED_NETS` für bestimmte Netze außer Kraft gesetzt werden. Hier können Netze angegeben werden, die *vertrauenswürdig* sind. Ein typisches Beispiel ist das Routing von NetBios-Ports zwischen zwei LANs, die über zwei Netzwerkkarten des fli4l-Routers versorgt werden. In diesem Fall sind alle vertrauenswürdigen Netze anzugeben, z.B.

```
TRUSTED_NETS='192.168.1.0/24 192.168.6.0/24'
```

Dabei ist es im Gegensatz zur Variable `MASQ_NETWORK` notwendig, alle Netze anzugeben, zwischen denen geforwardet werden soll. Es sind also mindestens 2 Netze einzutragen, damit korrekte IP-Tables-Regeln generiert werden!

Der Weg, den Pakete eines `ROUTE_NETWORK` nehmen, ist in Abbildung 3.4 dargestellt (es ist der gleiche wie ein Paket aus einem `TRUSTED_NET`).

Standard-Einstellung: `TRUSTED_NETS`="

Andere Netze Wie in den einführenden Bemerkungen bereits angemerkt, leitet der Router nur Pakete weiter, die aus Netzen kommen, die er kennt. Da man aber nicht alle Netze, mit denen man kommunizieren will, von vornherein kennt, führt der Router Buch über aufgebaute Verbindungen und lässt auch Pakete durch, die zu solchen Verbindungen gehören. Der Weg eines solchen Paketes durch den Paketfilter ist in Abbildung 3.5 dargestellt.

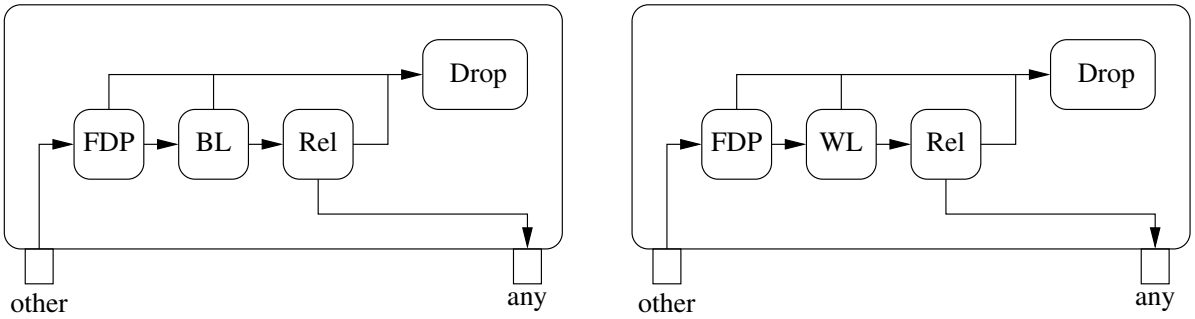


Abbildung 3.5.: Der Weg von Paketen aus einem unbekannten Netz durch den Paketfilter

3.12.2. Kommunikation mit dem Router

Während die bisherige Konfiguration die Kommunikation von Hosts über den Router hinweg beschrieb, beschreibt der folgende Teil die Kommunikation mit dem Router. Da fli4l Dienste auf bestimmte Ports zur Verfügung stellt, ist es sinnvoll, diese vor Zugriffen von aussen zu schützen.

Wichtig: *Das vom Router maskierte LAN (s.o) ist generell vor dem Zugang von aussen geschützt. Die angegebenen Portnummern regeln deshalb nur den Zugriff von aussen auf den fli4l-Router selbst.*

Interessant sind dabei die in Tabelle 3.5 beschriebenen Dienste und die dazugehörigen Ports. Rechner aus Netzen, die in `MASQ_NETWORK`, `ROUTE_NETWORK` bzw. `TRUSTED_NETS` angegeben sind, haben freien Zugang zum Router, für alle anderen sind alle Ports des Routers zu. Will man einen Dienst auch anderen anbieten, muss der entsprechende Port über die Variablen `INPUT_ACCEPT_PORT_x` geöffnet werden.

Port	Name	Prozess
22	ssh	sshd
37	time	<kernel>
53	domain	dnsmasq
80	www	mini_httpd
137:139	netbios	smbd (samba)
515	printer	lpd
5000	<imond>	imond
5001	<telmond>	telmond
8000	<proxy>	privoxy

Tabelle 3.5.: Ports, auf denen fli4l ggfs. Dienste anbietet

Es wird **dringend** empfohlen, die Standardkonfiguration der Firewall-Ports nicht zu verändern.

INPUT_ACCEPT_PORT_N INPUT_ACCEPT_PORT_x Standardmäßig sind alle Ports nach außen zu, eine Verbindung zum Router ist nicht möglich. Ports, auf denen ein Dienst nach außen angeboten wird, müssen hier angegeben werden.

Ein Eintrag besteht dabei aus einem einzelnem Port bzw. einem ganzen Port-Bereich (zwei Ports durch Doppelpunkt getrennt) und einer optionalen Protokollangabe (TCP oder UDP). Wird kein Protokoll angegeben, werden auf dem Port beide Protokolle akzeptiert. Am Beispiel der ssh, die ihren Dienst auf Port 22 über das TCP-Protokoll anbietet, würde das wie folgt aussehen:

```
INPUT_ACCEPT_PORT_N='1'              # no. of ports to accept from outside
INPUT_ACCEPT_PORT_1='22 TCP'         # e.g. allow connection to ssh service
```

INPUT_POLICY Kommt ein Paket beim Router an, das aufgrund der Konfiguration nicht durchgelassen werden darf, hat der Router zwei Möglichkeiten. Er kann das Paket einfach wegwerfen (*DROP*) oder es ablehnen (*REJECT*) und eine entsprechende Information an den Absender zurückschicken. Wie der Router reagiert, wird hier festgelegt, wobei die Standardreaktion *REJECT* ist.

Bei Anwendung der *DROP*-Policy verhält sich der Router still – das kann auch Probleme machen. Zum Beispiel schicken einige Rechner, die Internet-Dienste wie ftp anbieten, als Antwort auf einen Verbindungsaufbau eine Anfrage auf Port 113 (auth) zurück. Wenn fli4l darauf nicht reagiert, kann es zum ungewollten Abbruch der Verbindung zum gewünschen Dienst kommen.

Ich persönlich ziehe die *REJECT*-Policy vor, da sie erwartungsgemäß weniger Probleme bei diversen Internet-Protokollen bereitet und genauso sicher ist wie *DROP*.

Weiterführende Informationen zu *REJECT* vs *DROP*:

http://www.iks-jena.de/mitarb/lutz/usenet/Firewall.html#Deny
http://www.chiark.greenend.org.uk/~peterb/network/drop-vs-reject

DENY_ICMP Sollen Zugriffe von außen über das ICMP-Protokoll (echo requests) verboten werden, kann `DENY_ICMP='yes'` gesetzt werden. In diesem Fall kann man den Router von außen nicht mehr mit ping ansprechen. Bevor hier ein yes eingetragen wird, sollte man unbedingt folgende Seiten besuchen:

http://www.iks-jena.de/mitarb/lutz/usenet/Firewall.html#Verstecken
http://extern.fli4l.de/fli4l_faqengine/faq.php?display=faq&faqnr=26&catnr=10&prog=1

Standard-Einstellung: `DENY_ICMP='no'`

PACKETFILTER_LOG Hier kann eingestellt werden, ob abgelehnte Zugriffe von außen über die Syslog-Schnittstelle protokolliert werden sollen. Dazu muss auch `OPT_KLOGD` aktiviert sein.

Hier eine kurze Erklärung der wichtigsten Elemente einer Protokollzeile:

```
IN=ppp0 OUT= MAC= SRC=217.238.54.176 DST=217.235.38.43 LEN=48
    TOS=0x00 PREC=0x00 TTL=124 ID=30343 DF PROTO=TCP SPT=3087
    DPT=4662 WINDOW=16384 RES=0x00 SYN URGP=0
```

IN	Device, auf dem das Paket hereinkam
SRC, DST	Quell- und Zieladresse des Paketes
LEN	Länge des Pakets in Bytes
TOS	Type Of Service
TTL	Time To Live, Anzahl der Hops, bis das Paket gelöscht wird
PROTO=TCP	TCP Paket
SPT, DPT	Quell- und Zielport
ID	IP-ID, wird mit jedem Paket vom Sender um eins erhöht

Standard-Einstellung: `PACKETFILTER_LOG`='no'

PACKETFILTER_LOG_LEVEL Definiert die Log-Parameter, die beim Generieren eines Log-Eintrages verwendet werden. Möglich ist dabei eines der folgenden Log-Level: debug, info, notice, warning, err, crit, alert, emerg.

3.13. Neue Konfiguration des Paketfilters

Seit geraumer Zeit steht eine neue Möglichkeit zur Verfügung, den Paketfilter zu konfigurieren. Diese ist seit 2.1.8 die Standardkonfiguration. Die originale Konfiguration (Seite 41) existiert immer noch, muss aber separat aktiviert werden.

Mit der neuen Paketfilterkonfiguration lassen sich die einzelnen Tabellen des Paketfilters direkt konfigurieren. Dazu gibt es für jede relevante Tabelle eine eigene Liste, d.h. eine für die Input-Tabelle (`PF_INPUT_x`), eine für die Forwarding-Tabelle (`PF_FORWARD_x`) und eine für die Postrouting-Tabelle, in der das Masqerading gemacht wird (`PF_POSTROUTING_x`). Für die Prerouting-Tabelle gibt es momentan zwei Konfigurationsmöglichkeiten, die gleichzeitig genutzt werden können. Die bereits bekannte Portforwarding-Konfiguration über `PORTFW_x` (Seite 73) und `PF_PREROUTING_x`.

Ein Eintrag in einer dieser Listen, ausgenommen Pre-Routing, besteht im wesentlichen aus einer Aktion (*ACCEPT, DROP, REJECT, SNAT oder MASQUERADE*), die durch zusätzliche Bedingungen eingeschränkt werden kann. Diese Bedingungen beziehen sich auf Eigenschaften des betrachteten Paketes. Ein Paket enthält Informationen über seine Herkunft (Quelle, welcher Rechner hat das Paket losgeschickt), sein Ziel (an welchen Rechner und welche Anwendung soll das Paket gehen) u.a.m. Bedingungen können sich auf folgende Eigenschaften eines Paketes beziehen:

- die Quelle (Quell-IP, Quell-Port oder beides)
- das Ziel (Ziel-IP, Ziel-Port oder beides)
- das Protokoll
- das Interface, über welches das Paket reinkommt bzw. rausgeht
- die Mac-Adresse des Rechners, von dem das Paket kommt
- den Zustand des Paketes bzw. der Verbindung, zu der das Paket gehört

Kommt ein Paket rein, werden die Einträge bzw. die daraus generierten Regeln von oben nach unten abgearbeitet und die erste Aktion ausgeführt, bei der alle Bedingungen gelten. Trifft keine der Regeln zu, wird die Standardreaktion ausgeführt, die man für (fast) jede Tabelle angeben kann.

Ein Eintrag hat dabei folgendes Format, wobei zu beachten ist, dass alle Einschränkungen optional sind:

```
restriction{0,} [[source] [destination]] action [BIDIRECTIONAL|LOG|NOLOG]
```

3.13.1. Aktionen des Paketfilters

Aktionen können die folgenden sein:

ACCEPT Akzeptiere das Paket

DROP Verwerfe das Paket (der Absender erkennt das nur daran, dass keine Antwort, aber auch keine Fehlermeldung zurückkommt)

REJECT Weise das Paket zurück (der Absender enthält eine entsprechende Fehlermeldung)

LOG Logge das Paket und gehe zur nächsten Regel. Um verschiedene Log-Statements auseinanderhalten zu können, können Log-Prefixes verwendet werden. Diese werden via LOG:log-prefix angegeben.

MASQUERADE Maskiere das Paket (ersetze die Absender-IP des Paketes durch die eigene, bei der Einwahl dem Interface zugewiesene Adresse und sorge dafür, dass Antworten für diese Verbindung an den richtigen Rechner weitergeleitet werden; diese Aktion ist nur in `PF_POSTROUTING` möglich)

SNAT Ersetze die Absender-IP des Paketes durch die als Parameter für SNAT angegebene statische IP (für alle Pakete die zu der gerade betrachteten Verbindung gehören; diese Aktion ist nur in `PF_POSTROUTING` möglich)

NETMAP Mappe die Ziel- bzw. Absender-IP des Paketes in den als Parameter für NETMAP angegebenen Bereich, die Ports bleiben unverändert (für alle Pakete die zu der gerade betrachteten Verbindung gehören; diese Aktion ist nur in `PF_PREROUTING` oder `PF_POSTROUTING` möglich; in der Preroutinglist wird die Zieladresse verändert, in der Postroutinglist die Quell-Adresse)

DNAT Ersetze die Ziel-IP und den Zielport des Paketes durch die als Parameter für DNAT angegebene IP (für alle Pakete, die zu der gerade betrachteten Verbindung gehören; diese Aktion ist nur in `PF_PREROUTING` möglich)

REDIRECT Ersetze den Zielport des Paketes durch die als Parameter für REDIRECT angegebenen Port und stelle das Paket lokal zu (für alle Pakete, die zu der gerade betrachteten Verbindung gehören; diese Aktion ist nur in `PF_PREROUTING` möglich)

Einige dieser Aktionen können durch die Optionen *BIDIRECTIONAL*, *LOG* oder *NOLOG* in ihrem Verhalten modifiziert werden. *BIDIRECTIONAL* generiert die gleiche Regel noch einmal nur mit Quell- und Zieladresse vertauscht. *LOG/NOLOG* (de)aktivieren das Logging für diese eine Regel.

3.13.2. Einschränkungen in den Regeln

Einschränkungen können durch die in den folgenden Abschnitten aufgeführten Bedingungen vorgenommen werden. Bei den Bedingungen kann man immer 'any' angeben, wenn man an irgendeiner Stelle keine Einschränkung vornehmen will, aber trotzdem etwas angeben will/muss. Einschränkungen können in beliebiger Reihenfolge angegeben werden, wenn sie einen vorangestellten Präfix haben. Das gilt für alle außer die Angabe einer Quell- bzw. Zieladresse. Diese müssen immer direkt vor der Aktion stehen, die anderen Einschränkungen müssen vorher erfolgen. Einschränkungen können auch negiert werden, dazu wird einfach ein '!' vorangestellt.

Einschränkungen der Quelle und des Ziels

Jedes Paket enthält eine Quell- und eine Zielangabe; jeweils in Form eines Tupels einer IP-Adresse und eines Ports. Diese Quelle bzw. dieses Ziel kann für eine Einschränkung herangezogen werden. Die Angabe für die Quelle bzw. das Ziel kann folgendermaßen vorgenommen werden:

`ip`	eine einfache IP-Adresse
`network`	eine Netzwerkangabe der Form ip/netmask
`port[-port]`	ein Port bzw. ein Port-Bereich
`IP_NET_x_IPADDR`	die IP-Adresse des Interfaces x des Routers
`IP_NET_x`	das Subnetz x des Routers
`IP_ROUTE_x`	das in einer Route angegebene Subnetz (default Routen können nicht verwendet werden, sie würden any entsprechen und werden vorsichtshalber ausgeklammert)
`@name`	einer der via HOST_x_* vergebenen Namen oder Aliases; es wird die zugehörige IP-Adresse an dieser Stelle eingesetzt
`<ip oder netzwerk>:port[-port]`	IP/Netzwerk-Adresse in einer der obigen Varianten spezifiziert kombiniert mit einem Port/Port-Bereich

Das könnte z.B. wie folgt aussehen: `'192.168.6.2 any DROP'`

Tauchen zwei dieser Angaben auf, wird die erste als Quelle, die zweite als Ziel betrachtet. In diesem Beispiel verwerfen wir also Pakete, die vom Rechner mit der IP-Adresse 192.168.6.2 gesendet wurden, unabhängig davon, an welches Ziel sie gerichtet sind.

Taucht nur eine Angabe auf, wird anhand des Wertes entschieden, ob die Quelle oder das Ziel gemeint ist, wobei die Entscheidung relativ einfach ist:

- Ist eine Port-Angabe enthalten, ist das Ziel gemeint.
- Sonst ist die Quelle gemeint.

Wenn wir also z.B. das obige Beispiel etwas kürzer schreiben wollen, können wir einfach `'192.168.6.2 DROP'` schreiben. Es ist kein Port angegeben, die Bedingung gilt also für die Quelle, den Rechner, von dem das Paket gesendet wurde.

Wollen die Kommunikation mit dem sshd erlauben, können wir `'any any:22 ACCEPT'` (Pakete von einem beliebigen Rechner an einen beliebigen Rechner auf den sshd-Port 22 werden akzeptiert) oder kürzer `'22 ACCEPT'` schreiben. Es ist nur ein Port angegeben, also meinen wir das Ziel und damit Pakete, die an den Port 22 gerichtet sind.

Zur Vereinfachung der Regelmenge kann man an die Aktion ein *BIDIRECTIONAL* dranhängen, um auszudrücken, dass die Regel für beide Kommunikationsrichtungen gilt. Es werden

dann Regeln generiert, in denen einfach die Quell- und die Ziel-IP und eventuelle Interfaces vertauscht sind und der Rest gleichbleibt.
Beispiele:

`127.0.0.1 ACCEPT`	Lokale Kommunikation (Quelle 127.0.0.1) ist erlaubt
`any 192.168.12.1 DROP`	Pakete an die IP-Adresse 192.168.12.1 werden weggeworfen
`any 192.168.12.1 DROP LOG`	Pakete an die IP-Adresse 192.168.12.1 werden weggeworfen und zusätzlich gelogged
`any 192.168.12.1 DROP NOLOG`	Pakete an die IP-Adresse 192.168.12.1 werden weggeworfen, werden aber nicht gelogged
`22 ACCEPT`	Pakete an den Port 22 (sshd) werden akzeptiert
`IP_NET_1_NET ACCEPT`	Pakete aus dem am 1. Interface hängenden Subnetz werden akzeptiert
`IP_NET_1_NET IP_NET_2_NET`	Kommunikation zwischen dem am ersten und zweiten
`ACCEPT BIDIRECTIONAL`	Interface hängenden Subnetzen ist gestattet

Einschränkung des Interfaces

Einschränkung des Interfaces, über das das Paket reinkam bzw. rausgeht. Das Format sieht wie folgt aus: **if:***in***:***out*

In der Input-Liste kann man das rausgehende Interface nicht einschränken (das Paket geht ja nicht mehr raus), in der Masquerade-Liste (Postrouting) kann man das reinkommende Interface nicht einschränken, da die Information darüber nicht mehr vorhanden ist. Lediglich in der Forward-Liste kann man für beides Bedingungen angeben.

Möglich sind folgendes Werte für *in* bzw. *out*:

- lo (loopback Interface, lokale Kommunikation auf dem Router)
- circuit_1 – circuit_99
- `IP_NET_1_DEV` - `IP_NET_99_DEV`
- pppoe (das pppoe device)
- any

Einschränkungen des Protokolls

Einschränkung des Protokolls, zu dem dieses Paket gehört. Das Format sieht wie folgt aus: **prot:***protocol* bzw. **prot:***icmp***:***icmp-type*. *protocol* kann dabei einen der folgenden Werte annehmen:

- tcp
- udp
- icmp (Bei icmp kann man zusätzlich noch einen Namen für den icmp-Typ angeben, z. B. prot:icmp:echo-request)
- numerischer Wert der Protokoll-ID
- any

Einschränkung der MAC-Adresse

Mittels **mac:***mac addr* kann eine Einschränkung der MAC-Adresse vorgenommen werden.

Einschränkungen basierend auf dem Zustand eines Paketes

Der von fli4l verwendete Paketfilter sammelt Informationen über den Zustand von Verbindungen. Diese Informationen kann man dann nutzen, um Pakete zu Filtern, also z.B. nur Pakete durchzulassen, die zu bereits bestehenden Verbindungen gehören. Die Zustände einer Verbindung können ein:

INVALID Das Paket gehört zu keiner bekannten Verbindung.

ESTABLISHED Das Paket gehört zu einer Verbindung, über die bereits in beide Richtungen Pakete geflossen sind.

NEW Das Paket hat eine neue Verbindung aufgebaut oder gehört zu einer Verbindung, bei der noch nicht Pakete in beide Richtungen geflossen sind.

RELATED Das Paket baut eine neue Verbindung auf, hat aber eine Beziehung zu einer bestehenden Verbindung (z.B. ftp baut eine separate Verbindung für den eigentlich Datentransfer auf).

Für eine genauere Beschreibung der Zustände siehe
http://www.sns.ias.edu/~jns/files/iptables_talk/x38.htm.

Die Zustände werden wie folgt angegeben: **state:***state(s)*. Will man mehrere Zustände angeben, werden diese mit Komma getrennt. Um z.B. nur Pakete durchzulassen, die zu bestehenden Verbindungen gehören, kann man folgendes schreiben (für input und forwarding): **state:***ESTABLISHED,RELATED*

Einschränkung der Häufigkeit einer Aktion

Unter bestimmten Umständen möchte man die Häufigkeit von Aktionen begrenzen, z.B. nur einen *echo request* pro Sekunde zulassen. Das kann man mit der *limit* Einschränkung erreichen, die wie folgt aussieht: **limit:***Häufigkeit:Burst*. Die Häufigkeit wird dabei als n pro Zeiteinheit (second, minute, hour, day) angegeben wobei zusätzlich noch Ereignisse gehäuft auftreten können (Burst). *limit:3/minute:5* heißt z.B. 3 Ereignisse pro Minute, wobei auch mal 5 Ereignisse dicht aufeinander folgend akzeptiert werden.

3.13.3. Der Einsatz von Templates im Paketfilter

Um den Umgang mit dem Paketfilter weiter zu vereinfachen gibt es die Möglichkeit häufig vorkommende Regeln zu sogenannten Templates zusammenzufassen. Damit ist es möglich eine ganze Reihe von Paketfilterregeln zusammenzufassen und dieser Sammlung von Regeln einen symbolischen Namen zu geben. Anstatt direkt mit Protokollen und Portnummern arbeiten Sie dann mit Einträgen wie 'tmpl:ssh' wenn Sie mit dem SSH Protokoll arbeiten wollen. Wie mit Templates zu verfahren ist, wird hier am Beispiel von SSH gezeigt.

Wollen Sie Ihren fli4l-Router vom Internet aus per SSH erreichen, so schreiben Sie als Wert in der Variablen `PF_INPUT_x` den entsprechenden Dienstnamen mit vorangestelltem *tmpl:* und der Aktion, die für diesen Dienst gelten soll. Beispiel:

`PF_INPUT_2`='tmpl:ssh ACCEPT'

Hierbei steht *tmpl:* dafür, dass eine Regel aus einem Template erstellt werden soll. Den Namen des Dienstes geben Sie nach dem ':' an. In unserem Beispiel also 'ssh'. Zum Schluss geben

Sie an welche Aktion mit dem Dienst verbunden werden soll. Da Sie den fli4l-Router aus dem Internet erreichen wollen erlauben wir die Verbindung mit 'ACCEPT'. Einschränkungen von IP-Adressen oder Netzen sind nicht angegeben, also ist der SSH Dienst von allen Netzen und Interfaces erreichbar. Sie könnten bei Bedarf die gewohnten Schreibenweisen vom Paketfilter benutzen um den Zugriff auf den SSH Dienst weiter einzuschränken.

Für welche Dienste Regeln vorbereitet sind (=Templates existieren) kann in der Template Datei in `opt/etc/fwrules.tmpl/templates` nachgelesen werden, hier die Aufstellung in Tabellenform (siehe Tabelle 3.6).

Die Syntax für diese Form der Paketfilterregeln lautet also immer

tmpl:<Name des Dienstes> <Einschränkungen> <Gewünschte Aktion>

Wobei als <Einschränkungen> alles erlaubt ist was unter 3.13.2 beschrieben wird. Die möglichen Werte für <Gewünschte Aktion> sind in 3.13.1 aufgelistet und beschrieben.

Ein paar weitere Beispiele werden die Arbeitsweise verdeutlichen. Zuerst wollen wir uns `PF_PREROUTING` ansehen.

```
PF_PREROUTING_N='2'
PF_PREROUTING_1='tmpl:xbl dynamic DNAT:@xbox'
PF_PREROUTING_2='tmpl:https dynamic DNAT:192.168.193.250'
```

Die Regel `PF_PREROUTING_1` versorgt meine Xbox mit allem was für Xbox Live notwendig ist. Im einzelnen wird mit 'tmpl:xbl' alle Ports und Protokolle die für Xbox Live notwendig sind an den Host 'xbox' weitergeleitet. Anstelle der IP-Adresse wird ein Eintrag aus der `HOST_x_NAME` Liste benutzt. Durch 'dynamic' weiss der fli4l, dass die Ports vom Internetinterface weitergeleitet werden sollen.

Die zweite Regel leitet das https Protokoll an einen Webserver in der DMZ weiter.

Jetzt sehen wir uns an wie es mit `PF_INPUT` weitergeht.

```
PF_INPUT_N='3'
PF_INPUT_1='if:IP_NET_1_DEV:any ACCEPT'
PF_INPUT_2='if:pppoe:any prot:tcp 113 ACCEPT'
PF_INPUT_3='if:br0:any tmpl:dns @xbox 192.168.193.254 ACCEPT'
```

Die erste Regel lässt alle aus dem Netz das mit `IP_NET_1` definiert ist auf den Router zugreifen. Die zweite Regel ist für das oident Paket. Dort wird der ident Port geöffnet. Die dritte und letzte Regel erlaubt der Xbox den Zugriff auf den DNS Server auf dem fli4l. Hier ist auch wieder schon zu sehen wie man einen Hostalias einsetzt. Anstelle der IP-Adresse 192.168.193.254 hätte ich auch `IP_NET_1_IPADDR` schreiben können.

In `PF_FORWARD` und `PF_POSTROUTING` steht nichts 'tmpl:' spezifisches mehr.

Es ist auch möglich, dass Sie eigene Templates anlegen oder dass opt–Pakete ihre eigenen Templates mitbringen. Um ein eigenes Template anzulegen muss lediglich eine Datei mit den Namen des Templates erstellt werden und die entsprechenden Regeln dort aufgenommen werden. Wenn Sie eine private Templatedatei anlegen wollen erstellen Sie in dem Verzeichnis `etc/fwrules.tmpl` unterhalb Ihres config Verzeichnisses die Templatedatei, so wie es die Abbildung 3.6 zeigt. Wenn das Verzeichnis `etc/fwrules.tmpl` unterhalb Ihres config Verzeichnisses noch nicht existiert legen Sie bitte zuerst beide Verzeichnisse neu an. Alternativ können opt–Entwickler oder Benutzer die Templates für mehr als eine Konfiguration anlegen wollen Ihre Regeln direkt in das Verzeichnis `opt/etc/fwrules.tmpl` ablegen. In dieses Verzeichnis kommen dann die neuen Templates. Dabei gilt die Regel, dass die Templates im config Verzeichnis des Benutzers vor den Templates von opt–Paketen kommen. Zum Schluss wird die

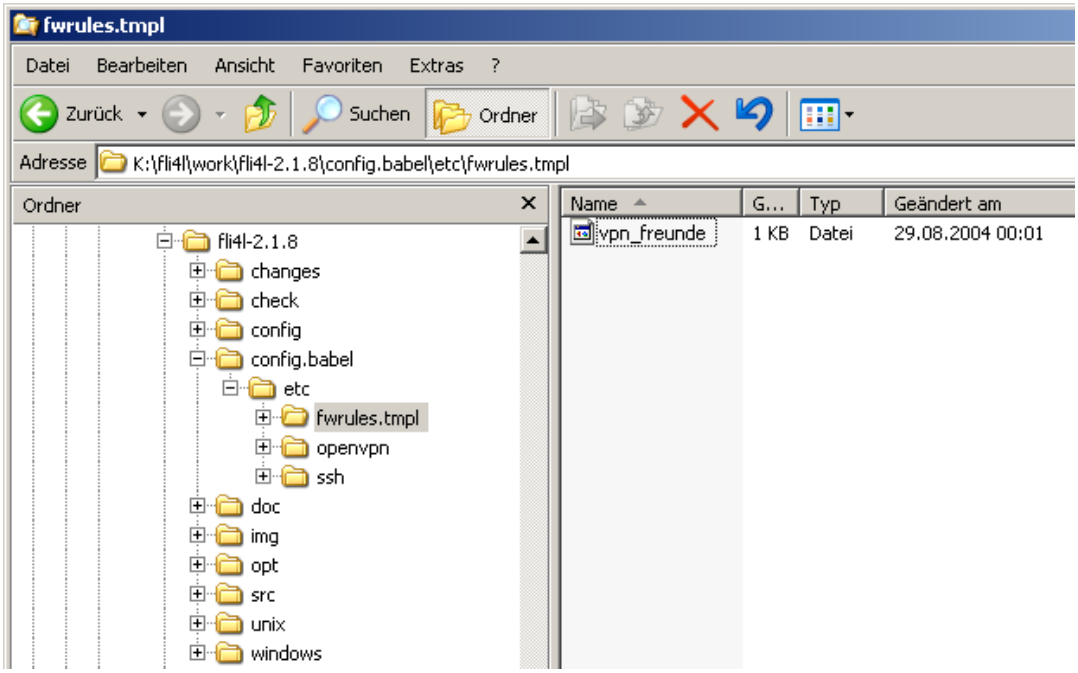


Abbildung 3.6.: Verzeichnisstruktur fli4l

templates Datei, die zum Lieferumfang von fli4l gehört, ausgewertet. Sie können also Einträge in der fli4l templates Datei dadurch „überschreiben“ indem Sie eine Templatedatei mit dem Namen des zu überschreibenden Templates in Ihrem config Verzeichnis anlegen.

Wenn Sie zum Beispiel das Template `vpn_freunde` anlegen wollen, legen Sie die Datei `vpn_freunde` an. Das Template soll die Dienste ssh, smtp, dns und samba enthalten. Also schreiben Sie in die Datei `vpn_freunde` folgendes:

```
prot:tcp 22
prot:tcp 25
53
prot:udp 137-138
prot:tcp 139
prot:tcp 445
```

Wann immer Sie jetzt das Template `vpn_freunde` benutzen, werden daraus Regeln für alle darin aufgeführten Protokolle und Ports erzeugt. `PF_FORWARD_x='tmpl:vpn_freunde ACCEPT'` erstellt folgende Regeln:

```
PF_FORWARD_x='prot:tcp 22 ACCEPT'
PF_FORWARD_x='prot:tcp 25 ACCEPT'
PF_FORWARD_x='53 ACCEPT'
PF_FORWARD_x='prot:udp 137-138 ACCEPT'
PF_FORWARD_x='prot:tcp 139 ACCEPT'
PF_FORWARD_x='prot:tcp 445 ACCEPT'
```

Templatename	Protokoll	Port(s)
dhcp	udp	67-68
dns	tcp/udp	53
elster	tcp	159.154.8.2:21
elster	tcp	159.154.8.35:21
elster	tcp	193.109.238.26:8000
elster	tcp	193.109.238.27:8000
elster	tcp	193.109.238.58:80
elster	tcp	193.109.238.59:80
elster	tcp	62.157.211.58:8000
elster	tcp	62.157.211.59:8000
elster	tcp	62.157.211.60:8000
elster	tcp	80.146.179.2:80
elster	tcp	80.146.179.3:80
ftp	tcp	21
http	tcp	80
https	tcp	443
hylafax	tcp	4559
imap	tcp	143
imaps	tcp	993
imond	tcp	5000
irc	tcp	6667
jabber	tcp	5222-5223
ldap	tcp/udp	389
mail	tcp	110
mail	tcp	143
mail	tcp	25
mail	tcp	465
mail	tcp	993
mail	tcp	995
mysql	tcp	3306
nntp	tcp	119
ntp	udp	123
oracle	tcp	1521
pcanywhere	tcp	5631-5632
ping	icmp:0	
ping	icmp:8	
pop3	tcp	110
pop3s	tcp	995
rdp	tcp	3389
rsync	tcp	873
samba	tcp	139
samba	tcp	445
samba	udp	137-138
smtp	tcp	25
snmp	tcp/udp	161
socks	tcp	1080
squid	tcp	3128
ssh	tcp	22
ssmtp	tcp	465
submission	tcp/udp	587
svn	tcp	3690
syslog	udp	514
teamspeak	tcp	14534
teamspeak	tcp	51234
teamspeak	udp	8767

Templatename	Protokoll	Port(s)
telmond	tcp	5001
telnet	tcp	23
teredo	udp	3544
time	tcp/udp	37
traceroute	udp	33404-33464
vnc	tcp	5900
whois	tcp	43
xbl	tcp/udp	3074
xbl	udp	88

Tabelle 3.6.: Die im Lieferumfang von fli4l enthaltenen Templates

3.13.4. Die Konfiguration des Paketfilters

Der Paketfilter im wesentlichen wird über 3 Listen konfiguriert:

- Input-Liste
- Forward-Liste
- Masquerade-Liste

Aktiviert wird die Konfiguration durch `PF_NEW_CONFIG`='yes'.

Für alle folgenden Listen gilt die in `PF_LOG_LEVEL` vorgenommene Einstellung des Log-Levels, dessen Inhalt auf einen der folgenden Werte gesetzt werden kann: debug, info, notice, warning, err, crit, alert, emerg.

Die Input-Liste

Über die Input-Liste wird konfiguriert, wer auf den Router zugreifen darf. Trifft keine der Regeln der Input-Liste zu, bestimmt die Default-Policy, was mit dem Paket passieren soll und die Log-Variable, ob es bei einer Ablehnung ins System-Log geschrieben werden soll.

Bei den verwendeten Parametern gibt es zwei Einschränkungen:

- *MASQUERADE* und *SNAT* können nicht als Aktion angegeben werden
- bei einer Interface-Einschränkung kann man nur das Eingangsinterface einschränken.

PF_INPUT_POLICY Diese Variable beschreibt die Default Policy, die angewand wird, wenn keine der anderen Regeln zutrifft. Möglich sind:

- ACCEPT (nicht empfohlen)
- REJECT
- DROP (nicht empfohlen)

PF_INPUT_ACCEPT_DEF Steht diese Variable auf 'yes', werden default-Regeln generiert, die für ein korrektes Funktionieren des Routers notwendig sind. Standardmäßig sollte man hier 'yes' eintragen.

Möchte man das Verhalten komplett selbst definieren, kann man hier 'no' eintragen, muss dann jedoch alle Regeln selbst definieren. Eine zum Standardverhalten äquivalente Konfiguration würde wie folgt aussehen (die Beschreibung der Liste nutzerdefinierter Chains erfolgt hier (Seite 60):

```
PF_INPUT_ACCEPT_DEF='no'
#
# limit icmp echo requests - use a seperate chain
#
PF_USR_CHAIN_N='1'
PF_USR_CHAIN_1_NAME='usr-in-icmp'
PF_USR_CHAIN_1_RULE_N='1'
PF_USR_CHAIN_1_RULE_1='prot:icmp:echo-request length:0-100 limit:1/second:5 ACCEPT'

PF_INPUT_N='4'
PF_INPUT_1='state:ESTABLISHED,RELATED ACCEPT'
PF_INPUT_2='prot:icmp usr-in-icmp'
PF_INPUT_3='if:lo:any ACCEPT'
PF_INPUT_4='state:NEW 127.0.0.1 DROP BIDIRECTIONAL'
```

Die erste Regel akzeptiert nur solche Pakete, die zu bestehenden Verbindungen gehören (`'state:ESTABLISHED,RELATED ACCEPT'`, damit auch alle icmp pakete, die Resultat einer bestehenden Verbindung sind), die zweite Regel verzweigt zur ratenlimitierenden usr-in-icmp chain und die dritte erlaubt lokale Kommunikation ('if:lo:any ACCEPT'). Die vierte filtert Pakete heraus, die behaupten, lokale Kommunikation zu sein, aber nicht bereits von der vorherigen Regel akzeptiert wurden.

Verwendet man eine DMZ oder arbeitet mit OpenVPN, muss man die Regeln noch ergänzen, um die von diesen Paketen verwendeteten chains einzubinden.

```
PF_INPUT_N='6'
...
PF_INPUT_5='dmz-chain'
PF_INPUT_6='ovpn-chain'
```

PF_INPUT_LOG Definiert, ob abgelehnte Pakete vom Kernel protokolliert werden sollen. Dabei können die Meldungen durch Aktivierung von KLOGD an den SYSLOGD über- und durch diesen, entsprechend der Konfiguration ausgegeben werden.

PF_INPUT_N PF_INPUT_x PF_INPUT_x_COMMENT Liste der Regeln, die beschreiben, welche Pakete vom Router angenommen bzw. verworfen werden.

PF_INPUT_LOG_LIMIT Definiert, wie häufig Log-Einträge generiert werden, die Häufigkeit wird analog zur Limit-Einschränkung als n pro Zeiteinheit mit Bursts beschrieben, also z.B. '3/minute:5'. Ist dieser Eintrag leer, wird der Standardwert '1/second:5' verwendet, enhält er 'none', wird keine Limitierung durchgeführt.

PF_INPUT_REJ_LIMIT PF_INPUT_UDP_REJ_LIMIT Definiert, wie häufig bei einer Ablehnung eines reinkommenden Paketes auch ein entsprechendes Reject-Paket generiert wird, die Häufigkeit wird analog zur Limit-Einschränkung als n pro Zeiteinheit mit Bursts beschrieben, also z.B. '3/minute:5'. Ist das Limit überschritten, wird das Paket einfach gedroppt. Ist dieser Eintrag leer, wird der Standardwert '1/second:5' verwendet, enhält er 'none', wird keine Limitierung durchgeführt.

PF_INPUT_ICMP_ECHO_REQ_LIMIT Definiert, wie häufig auf ein icmp echo request reagiert werden soll, die Häufigkeit wird analog zur Limit-Einschränkung als n pro Zeiteinheit mit Bursts beschrieben, also z.B. '3/minute:5'. Ist das Limit überschritten, wird das Paket einfach gedroppt. Ist dieser Eintrag leer, wird der Standardwert '1/second:5' verwendet, enhält er 'none', wird keine Limitierung durchgeführt.

Die Forward-Liste

Über die Forward-Liste wird konfiguriert, welche Pakete vom Router weitergeleitet werden. Trifft keine der Regeln der Forward-Liste zu, bestimmt die Default-Policy, was mit dem Paket passieren soll und die Log-Variable, ob es bei einer Ablehnung ins System-Log geschrieben werden soll.

Bei den verwendeten Parametern gibt es zwei Einschränkungen:

- *MASQUERADE* und *SNAT* können nicht als Aktion angegeben werden.
- *REJECT* kann (momentan noch) nicht als Aktion angegeben werden.

PF_FORWARD_POLICY Die Default Policy wird angewand, wenn keine der Regeln zutrifft. Möglich sind:

- ACCEPT
- DROP

PF_FORWARD_ACCEPT_DEF Bestimmt, ob der Router Pakete akzeptiert, die zu bestehenden Verbindungen gehören. Steht diese Variable auf 'yes', generiert fli4l automatisch eine Regel, die Pakete mit dem entsprechenden Zustand akzeptiert

`'state:ESTABLISHED,RELATED ACCEPT'`,

sowie eine Regel, die Pakete mit gefälschten IP-Adressen verwirft

`'state:NEW 127.0.0.1 DROP BIDIRECTIONAL'`.

Zusätzlich generieren die anderen Subsysteme auch noch Standardregeln – eine Konfiguration ohne Standardregeln mit Portforwarding, DMZ und OpenVPN würde mindestens folgende Regeln enthalten:

```
PF_FORWARD_ACCEPT_DEF='no'
PF_FORWARD_N='5'
PF_FORWARD_1='state:ESTABLISHED,RELATED ACCEPT'
PF_FORWARD_2='state:NEW 127.0.0.1 DROP BIDIRECTIONAL'
PF_FORWARD_3='pfwaccess-chain'
PF_FORWARD_4='dmz-chain'
PF_FORWARD_5='ovpn-chain'
```

PF_FORWARD_LOG Definiert, ob abgelehnte Pakete vom Kernel protokolliert werden sollen. Dabei können die Meldungen durch Aktivierung von KLOGD an den SYSLOGD über- und durch diesen, entsprechend der Konfiguration ausgegeben werden.

PF_FORWARD_LOG_LIMIT Definiert, wie häufig Log-Einträge generiert werden, die Häufigkeit wird analog zur Limit-Einschränkung als n pro Zeiteinheit mit Bursts beschrieben, also z.B. '3/minute:5'. Ist dieser Eintrag leer, wird der Standardwert '1/second:5' verwendet, enhält er 'none', wird keine Limitierung durchgeführt.

PF_FORWARD_REJ_LIMIT PF_FORWARD_UDP_REJ_LIMIT Hiermit wird definiert, wie häufig bei einer Ablehnung eines reinkommenden Paketes auch ein entsprechendes Reject-Paket generiert wird, die Häufigkeit wird analog zur Limit-Einschränkung als n pro Zeiteinheit mit Bursts beschrieben, also z.B. '3/minute:5'. Ist das Limit überschritten, wird das Paket einfach gedroppt. Ist dieser Eintrag leer, wird der Standardwert '1/second:5' verwendet, enhält er 'none', wird keine Limitierung durchgeführt.

PF_FORWARD_N PF_FORWARD_x PF_FORWARD_x_COMMENT Liste der Regeln, die beschreiben, welche Pakete vom Router weitergeleitet werden bzw. verworfen werden.

Nutzerdefinierte Listen

Aus verschiedenen Gründen besteht manchmal der Bedarf, eigene Chains anzulegen und dort die Pakete genauer zu filtern. Diese Chains kann man mittels `PF_USR_CHAIN_%` definieren und füllen. Die Chain-Namen müssen dabei mit *usr-* beginnen und können nach ihrer Definition überall in der Input oder Forward-Liste statt einer Aktion eingesetzt werden. Als Beispiel soll hier die bereits vorher verwendete icmp-Filterchain dienen:

```
PF_USR_CHAIN_N='1'
#
# create usr-in-icmp
#
PF_USR_CHAIN_1_NAME='usr-in-icmp'
#
# add rule to usr-in-icmp
#
PF_USR_CHAIN_1_RULE_N='1'
PF_USR_CHAIN_1_RULE_1='prot:icmp:echo-request length:0-100 limit:1/second:5 ACCEPT'
#
# use chain in PF_INPUT
#
PF_INPUT_2='prot:icmp usr-in-icmp'
```

PF_USR_CHAIN_N Anzahl der Nutzerdefinierten Chains

PF_USR_CHAIN_x_NAME Definiert den Namen der Nutzerdefinierten Chain. Dieser muss mit *usr-* beginnen.

PF_USR_CHAIN_x_RULE_N

PF_USR_CHAIN_x_RULE_x

PF_USR_CHAIN_x_RULE_x_COMMENT Hier werden die Regeln definiert, die in die userdefinierte Liste eingefügt werden sollen. Es können alle Regeln verwendet werden, die auch in einer Forward-Liste verwendet werden könnten. Sollte keine Regel der User-Chain zutreffen, wird zur Ausgangs-Chain zurückgekehrt und mit der Regel nach der Verzweigung fortgefahren.

Die NAT-Listen (network address translation)

Pakete können vor und nach Routing-Entscheidungen noch manipuliert werden, sie können zum Beispiel eine neue Zieladresse erhalten, um sie an einen anderen Rechner weiterzuleiten (Portforwarding) oder eine andere Quell-Adresse erhalten, um das hinter dem Router liegende Netzwerk zu maskieren. Maskieren nutzt man z.B. um ein privates Netz über eine öffentliche IP ins Netz zu bringen oder in einem DMZ Setup die Struktur des lokalen Netzes vor den Rechnern in der DMZ zu verbergen.

Die Konfiguration erfolgt über zwei Listen, die pre- und die postrouting-Liste.

Über die Prerouting-Liste wird konfiguriert, welche Pakete an einen anderen Rechner weitergeleitet werden sollen. Trifft keine der Regeln der Prerouting-Liste zu, werden die Pakete unverändert weiterbehandelt.

Beim Maskieren gibt es zwei Varianten, eine für Interfaces, die bei der Einwahl erst eine IP zugewiesen bekommen (*MASQUERADE*) und eine für Interfaces mit statischer IP (*SNAT*). Letzteres erwartet dabei zusätzliche Parameter:

- die IP, die im Paket als Quelle eingetragen werden soll. Diese kann als
  - IP-Adresse (SNAT:1.2.3.4),
  - IP-Bereich (SNAT:1.2.3.4-1.2.3.10),
  - oder als Interface-Name angegeben werden, dessen IP-Adresse als Quelle dienen soll (SNAT:ip:eth_x_ipaddr).

Bei beiden kann ein Port/Portbereich angegeben werden, auf den der Quellport gemapped werden soll (Normalerweise ist das nicht nötig, da der Kern die Ports allein auswählen kann. Es gibt aber Anwendungen, die verlangen, dass der Quellport unverändert bleibt (erfordern ein 1:1 NAT oder verbieten PAT (Port Address Translation) bzw. NAPT (Network Port Address Translation)). Der Portbereich wird einfach hinten angehängt, z.B. so: SNAT:IP_NET_1_IPADDR:4000-8000.

Bei den verwendeten Parametern gibt es zwei Einschränkungen:

- *DROP* kann nicht als Aktion angegeben werden.
- *REJECT* kann nicht als Aktion angegeben werden.

PF_POSTROUTING_N PF_POSTROUTING_x PF_POSTROUTING_x_COMMENT
Eine Liste der Regeln, die beschreiben, welche Pakete vom Router maskiert werden (bzw. unmaskiert weitergeleitet werden). Will man Pakete vom Maskieren ausklammern, kann man eine Accept-Regel für die auszuklammernden Pakete der Masquerade-Regel voranstellen.

Über die Postrouting-Liste wird konfiguriert, welche Pakete vom Router maskiert werden. Trifft keine der Regeln der Postrouting-Liste zu, werden die Pakete unmaskiert weitergeleitet.

DNAT erwartet als Parameter eine IP, die als neue Adresse des Paketes eingetragen werden soll und akzeptiert auch einen neuen Port:

- IP-Adresse: DNAT:1.2.3.4
- IP-Adresse und neuer Port: DNAT:1.2.3.4:8080
- IP-Adresse als Name: DNAT:client1

REDIRECT verhält sich wie DNAT, nur dass die Ziel-IP-Adresse immer auf 127.0.0.1 gesetzt wird und damit das Paket lokal zugestellt wird. Dies wird z.B. für transparente Proxys benötigt, siehe OPT_TRANSPROXY (Seite 218).

Will man Portforwarding auf Interfaces mit dynamischen IPs machen, weiß man zum Zeitpunkt der Konfiguration noch nicht, an welche IP die Pakete gerichtet sein werden. Daher kann man in der Prerouting-Liste *dynamic* als Platzhalter für die später zugewiesene IP verwenden. z.B. wie folgt:

```
'dynamic:80  DNAT:1.2.3.4'           # forwarde http-Pakete
                                     # an die ip 1.2.3.4
'prot:gre any dynamic DNAT:1.2.3.4'  # forwarde gre (teil des pptp
                                     # protokolls) an die ip 1.2.3.4
```

Bei den verwendeten Parametern gibt es eine Einschränkung, *REJECT* kann nicht als Target angegeben werden.

PF_PREROUTING_N PF_PREROUTING_x PF_PREROUTING_x_COMMENT
: Regeln, die beschreiben, welche Pakete vom Router an ein anderes Ziel weitergeleitet werden sollen.

3.13.5. Beispiele

Im Folgenden einige Beispiele für die neue Konfiguration.

Die Fli4l-Standardkonfiguration

Die fli4l-Standardkonfiguration der Distribution sah wie folgt aus, wenn man alle deaktivierten Optionen weglässt:

```
MASQ_NETWORK='192.168.6.0/24' # networks to masquerade (e.g. our LAN)
PF_INPUT_POLICY='REJECT'         # policy for input chain: reject or drop
FORWARD_DENY_PORT_N='1'       # no. of ports to reject/deny forwarding
FORWARD_DENY_PORT_1='137:139 reject'  # drop/reject forwarding of netbios
DENY_ICMP='no'                # deny icmp (ping): yes or no
PACKETFILTER_LOG='no'         # log access to rejected/denied ports
```

Nehmen wir diese Optionen und die im Paketfilter-Kapitel beschriebenen Annahmen, sieht eine Übersetzung ins neue Schema wie folgt aus:

```
PF_INPUT_POLICY='REJECT'
PF_INPUT_ACCEPT_DEF='yes'
PF_INPUT_LOG='no'
PF_INPUT_N='1'
PF_INPUT_1='IP_NET_1 ACCEPT'
```

Damit erreichen wir, dass

- Rechner im lokalen Netzwerk auf den Router zugreifen dürfen (Regel 1),
- lokale Kommunikation auf dem Router erlaubt ist (`PF_INPUT_ACCEPT_DEF`),
- Pakete, die zu vom Router aufgebauten Verbindungen gehören, akzeptiert werden (`PF_INPUT_ACCEPT_DEF`),
- alles andere abgelehnt wird (`PF_INPUT_POLICY`),
- aber nicht ins Systemlog geschrieben wird.

Für die Forward-Liste sieht das so ähnlich aus. Nur Pakete unseres lokalen Netzes und Pakete, die zu Verbindungen gehören, die von Rechnern im lokalen Netz aufgebaut wurden, sollen weitergeleitet werden. Netbios-Pakete und der neue cifs Port werden verworfen.

```
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'
PF_FORWARD_N='2'
PF_FORWARD_1='tmpl:samba DROP'
PF_FORWARD_2='IP_NET_1 ACCEPT'
```

Was man hier gut sieht, ist die Abhängigkeit von der Reihenfolge der Regeln. Als erstes werden Netbios-Pakete verworfen, dann werden die Pakete des lokalen Netzes akzeptiert.

Nun kann das lokale Netz mit dem Router kommunizieren, seine Pakete werden weitergeleitet, es fehlt nur noch das Maskieren, welches für den Zugriff eines privaten Netzwerkes auf das Internet notwendig ist:

```
PF_POSTROUTING_N='1'
PF_POSTROUTING_1='IP_NET_1  MASQUERADE'
```

Für diesen einfachen Fall sieht die ursprüngliche Konfiguration wesentlich einfacher aus. Das ändert sich, wenn wir die anderen Optionen mit hinzunehmen, die da sind:

- trusted nets
- route network
- white list
- black list

Trusted Nets

Wollen wir lokal mehrere Subnetze haben, die frei und unmaskiert miteinander kommunizieren können, müssen wir dasfür sorgen, dass Pakete zwischen diesen Subnetzen nicht verworfen und auch nicht maskiert werden. Dazu wurden in der originalen FW-Konfiguration diese Netze in `TRUSTED_NETS` aufgezählt. In der neuen Konfiguration fügen wir einfach eine Regel hinzu oder modifizieren die vorhandene.

Angenommen, wir haben einen DSL-Zugang über pppoe und die beiden Subnetze sind `IP_NET_1` (192.168.6.0/24) und `IP_NET_2` (192.168.7.0/24). Dann würde die Konfiguration wie folgt aussehen:

```
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'
PF_FORWARD_N='4'
PF_FORWARD_1='IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_FORWARD_2='tmpl:samba DROP'
PF_FORWARD_3='IP_NET_1 ACCEPT'
PF_FORWARD_4='IP_NET_2 ACCEPT'

PF_POSTROUTING_N='3'
PF_POSTROUTING_1='IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_POSTROUTING_2='IP_NET_1 MASQUERADE'
PF_POSTROUTING_3='IP_NET_2 MASQUERADE'
```

Die Regel eins sorgt jetzt dafür, dass Pakete zwichen den beiden Subnetzen ohne weitere Prüfung weitergeleitet werden. Regel 3 und 4 sorgen dafür, dass beide Subnetze auch ins Internet kommen. Die erste Regel der MASQ-Liste sorgt dafür, dass die Kommunikation zwischen den Subnetzen unmaskiert erfolgt. Alternativ könnten wir auch sagen, dass nur Pakete, die über das pppoe-Device rausgehen, maskiert werden sollen:

```
PF_POSTROUTING_N='1'
PF_POSTROUTING_1='if:any:pppoe MASQUERADE'
```

Genauso hätte man die Filterung der Ports auch auf das pppoe-device beschränken und die beiden Subnetze zu einem zusammenfassen können, das würde dann wie folgt aussehen:

```
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'
PF_FORWARD_N='2'
PF_FORWARD_1='if:any:pppoe tmpl:samba DROP'
PF_FORWARD_2='192.168.6.0/23 ACCEPT'

PF_POSTROUTING_N='1'
PF_POSTROUTING_1='if:any:pppoe MASQUERADE'
```

Pakete, die über das pppoe-Device hinaus gehen und die an die Ports 137-138 udp sowie 139 und 445 tcp adressiert sind, werden verworfen (Regel 1), alle anderen Pakete, die aus dem Subnetz 192.168.6.0/23 kommen, werden weitergeleitet (Regel 2).

Route Network

Fügen wir dem ganzen noch ein Netzwerk 10.0.0.0/24 hinzu (z.B. ein Dial-In-Netzwerk, mit dem wir unmaskiert kommunizieren wollen, wobei die Ports 137-138 udp, 139 und 445 tcp verworfen werden sollen. Das würde wie folgt aussehen:

```
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'
PF_FORWARD_N='4'
PF_FORWARD_1='IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_FORWARD_2='tmpl:samba DROP'
PF_FORWARD_3='192.168.6.0/23 ACCEPT'
PF_FORWARD_4='10.0.0.0/24 ACCEPT'

PF_POSTROUTING_N='2'
PF_POSTROUTING_1='10.0.0.0/24 ACCEPT BIDIRECTIONAL'
PF_POSTROUTING_2='192.168.6.0/23 MASQUERADE'
```

- Regel 1 erlaubt die ungehinderte Kommunikation zwischen den Subnetzen `IP_NET_1` und `IP_NET_2`
- Regel 2 verwirft die Samba-Ports
- Regel 3 und 4 lassen Pakete durch, die aus den Subnetzen 192.168.6.0/24, 192.168.7.0/24 und 10.0.0.0/24 kommen, die Rückrichtung wird von `FORWARD_ACCEPT_DEF`='yes' abgedeckt.
- Regel 1 der Masq-Liste sorgt dafür, dass Pakete in das/aus dem 10.0.0.0/24 Subnetz nicht maskiert werden.

Alternativ ginge auch:

```
PF_POSTROUTING_N='1'
PF_POSTROUTING_1='if:any:pppoe MASQUERADE'
```

Nur Pakete, die über das pppoe-Interface hinausgehen, werden maskiert.

Blacklists, Whitelists

Blacklists (ein Rechner in dieser Liste darf etwas nicht) und Whitelists (ein Rechner in dieser Liste darf etwas) werden prinzipiell ähnlich umgesetzt. Es werden Regeln geschrieben, die am Anfang sehr speziell sind und nach hinten immer allgemeiner werden. Bei einer Blacklist stehen am Anfang Regeln, die etwas verbieten und am Ende Regeln, die allen bisher nicht erwähnten etwas erlauben, bei einer Whitelist ist es genau umgekehrt. Einfaches Beispiel: Alle Rechner im Subnetz 192.168.6.0/24 außer Rechner 12 dürfen ins Internet, solange sie nicht mit den CIFS Ports 137-138 udp, 139 und 445 tcp kommunizieren wollen:

```
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'
PF_FORWARD_N='3'
```

```
PF_FORWARD_1='192.168.6.12 DROP'
PF_FORWARD_2='tmpl:samba DROP'
PF_FORWARD_3='192.168.6.0/23 ACCEPT'

PF_POSTROUTING_N='1'
PF_POSTROUTING_2='192.168.6.0/24 MASQUERADE'
```

Nur Rechner 12 darf ins Internet (aber nicht an die Ports ...), alle anderen dürfen nur lokal mit einem anderen Subnetz kommunizieren:

```
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'
PF_FORWARD_N='3'
PF_FORWARD_1='192.168.6.0/24 192.168.7.0/24 ACCEPT BIDIRECTIONAL'
PF_FORWARD_2='tmpl:samba DROP'
PF_FORWARD_3='192.168.6.12 ACCEPT'

PF_POSTROUTING_N='1'
PF_POSTROUTING_1='if:any:pppoe MASQUERADE'
```

3.13.6. Standardkonfigurationen

Einfacher maskierender Router mit einem Netz dahinter

```
PF_NEW_CONFIG='yes'

#
# Zugriff auf den Router
#
PF_INPUT_POLICY='REJECT'
PF_INPUT_ACCEPT_DEF='yes'
PF_INPUT_LOG='no'
PF_INPUT_N='1'
PF_INPUT_1='IP_NET_1 ACCEPT'   # alle hosts im lokalen netz dürfen
 # auf den router zugreifen

#
# Zugriff auf das ``Internet''
#
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'

PF_FORWARD_N='2'
PF_FORWARD_1='tmpl:samba DROP' # samba pakete, die das netz
 # verlassen wollen, werden verworfen
PF_FORWARD_2='IP_NET_1 ACCEPT' # alle anderen pakete dürfen das
 # lokale netz verlassen

#
# Maskieren des lokalen Netzes
#
```

```
PF_POSTROUTING_N='1'
PF_POSTROUTING_1='IP_NET_1 MASQUERADE'  # masquerade traffic leaving
                                # the subnet
```

Einfacher maskierender Router mit zwei Netzen dahinter

```
PF_NEW_CONFIG='yes'

#
# Zugriff auf den Router
#
PF_INPUT_POLICY='REJECT'
PF_INPUT_ACCEPT_DEF='yes'
PF_INPUT_LOG='no'
PF_INPUT_N='2'
PF_INPUT_1='IP_NET_1 ACCEPT'   # alle hosts im lokalen netz dürfen
 # auf den router zugreifen
PF_INPUT_2='IP_NET_2 ACCEPT'   # alle hosts im lokalen netz dürfen
 # auf den router zugreifen

#
# Zugriff auf das ``Internet''
#
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'

#
# Kein Samba zwischen den Netzen
#
PF_FORWARD_N='3'
PF_FORWARD_1='tmpl:samba DROP' # samba pakete, die das netz
 # verlassen wollen, werden verworfen
PF_FORWARD_2='IP_NET_1 ACCEPT' # alle anderen pakete dürfen das
 # lokale netz verlassen
PF_FORWARD_3='IP_NET_2 ACCEPT' # alle anderen pakete dürfen das
 # lokale netz verlassen

#
# Freie Kommunikation zwischen den Netzen
#
PF_FORWARD_N='4'
PF_FORWARD_1='IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_FORWARD_2='tmpl:samba DROP' # samba pakete, die das netz
 # verlassen wollen, werden verworfen
PF_FORWARD_3='IP_NET_1 ACCEPT' # alle anderen pakete dürfen das
 # lokale netz verlassen
PF_FORWARD_4='IP_NET_2 ACCEPT' # alle anderen pakete dürfen das
 # lokale netz verlassen

#
# Maskieren der lokalen Netze, unmaskierte Kommunikation zwischen den
# Netzen
```

```
#
PF_POSTROUTING_N='3'
PF_POSTROUTING_1'IP_NET_1 IP_NET_2 ACCEPT BIDIRECTIONAL'
PF_POSTROUTING_2='IP_NET_1 MASQUERADE'  # masquerade traffic leaving
                                        # the subnet
PF_POSTROUTING_3='IP_NET_2 MASQUERADE'  # masquerade traffic leaving
                                        # the subnet
```

Maskierender DSL-Router mit zwei Netzen dahinter und ssh/http-Zugriff aus dem Internet

```
PF_NEW_CONFIG='yes'

#
# Zugriff auf den Router
#
PF_INPUT_POLICY='REJECT'
PF_INPUT_ACCEPT_DEF='yes'
PF_INPUT_LOG='no'
PF_INPUT_N='4'
PF_INPUT_1='IP_NET_1 ACCEPT'   # alle hosts im lokalen netz dürfen
 # auf den router zugreifen
PF_INPUT_2='IP_NET_2 ACCEPT'   # alle hosts im lokalen netz dürfen
 # auf den router zugreifen
PF_INPUT_3='tmpl:ssh ACCEPT'   # gestatte Zugriff auf ssh-Service
 # von Überall her
PF_INPUT_4='tmpl:http 1.2.3.4/24 ACCEPT'  # gestatte Rechner aus
 # einem bestimmten Subnetz Zugriff
 # auf http-Service


#
# Zugriff auf das ``Internet''
#
PF_FORWARD_POLICY='REJECT'
PF_FORWARD_ACCEPT_DEF='yes'
PF_FORWARD_LOG='no'

#
# Keine Kommunikation zwischen den Netzen, beide Netze dürfen ins
# Internet, Samba-Pakete werden verworfen
#
PF_FORWARD_N='2'
PF_FORWARD_1='tmpl:samba if:any:pppoe DROP' # samba pakete, die das netz
     # verlassen wollen, werden verworfen
PF_FORWARD_2='if:any:pppoe ACCEPT' # alle anderen pakete dürfen das
     # lokale netz verlassen

#
# Maskieren der lokalen Netze, unmaskierte Kommunikation zwischen den
# Netzen
#
PF_POSTROUTING_N='1'
```

```
PF_POSTROUTING_1='if:any:pppoe MASQUERADE'  # masquerade traffic leaving
                                            # the subnet
```

Portforwarding

Will man statt der alten Portforwarding-Konfiguration die neue verwenden, lässt sich das wie folgt übersetzen:

```
TARGET='<port>'
NEW_TARGET='<ip>'
PROTOCOL='<proto>'
PF_PREROUTING_x='prot:<proto> dynamic:<port> DNAT:<ip>'

TARGET='<port1>-<port2>'
NEW_TARGET='<ip>'
PROTOCOL='<proto>'
PF_PREROUTING_x='prot:<proto> dynamic:<port1>-<port2> DNAT:<ip>'

TARGET='<ip>:<port-a>'
NEW_TARGET='<ip>:<port-b>'
PROTOCOL='<proto>'
PF_PREROUTING_x='prot:<proto> any <ip>:<port-a> DNAT:<ip>:<port-b>'
```

Transparent Proxy

Will man bestimmte Zugriffe auf das Internet nur über einen lokalen Proxy zulassen, kann man das mit Hilfe der Pre/Postroutinglisten erzwingen, ohne dass der Client davon etwas merkt.

Prinzipiell sind dazu drei Schritte notwendig:

1. Anfragen an den http Port, die nicht vom proxy kommen, an den Proxy umleiten (PREROUTING).
2. Die umgeleiteten Pakete so verändern, dass der Proxy denkt, sie kommen vom Router, so dass er sie wieder dorthin zurückschickt (POSTROUTING).
3. Die Pakete durch die Forwardchain durchlassen, sofern ein Eintrag alà

   `PF_FORWARD_x`='IP_NET_1 ACCEPT'

   nicht existiert (FORWARD).

1. Beispiel: Angenommen wir haben nur ein Netz `IP_NET_1`, in dem auf einem Rechner namens "proxy" ein Squid läuft und wollen den gesamten http-Traffic über ihn leiten. Squid lauscht auf Port 3128. Der Einfachheit halber beziehen wir uns via '@proxy' auf den eingetragenen Host aus `HOST_1_NAME`='proxy' (vgl. Domainkonfiguration (Seite 76)).

Das ganze würde wie folgt aussehen:

```
...
  PF_PREROUTING_x='@proxy ACCEPT'
      # Pakete vom Proxy sollen nicht umgeleitet werden

  PF_PREROUTING_x='prot:tcp IP_NET_1 any:80 DNAT:@proxy:3128'
      # http-Pakete aus IP_NET_1 mit einem beliebigen Ziel werden
```

```
      # umgeleitet nach @proxy Port 3128

  PF_POSTROUTING_x='any @proxy:3128 SNAT:IP_NET_1_IPADDR'
      # alle Pakete an den Proxy Port 3128 so umschreiben als wären sie
      # vom fli4l (IP_NET_1_IPADDR)

  PF_FORWARD_x='prot:tcp @proxy 80 ACCEPT'
      # http-Pakete vom Proxy durch die Forwardchain durchlassen (wenn nötig)
...
```

Gibt es mehrere Netze oder potentielle Konflikte mit anderen Portforwardings (die ja auch nichts anderes sind als DNAT-Regeln), muss man die Regeln vielleicht noch etwas enger formulieren.

2. Beispiel: Unser Proxy namens "proxy" steht in `IP_NET_1`, lauscht auf Port 3128 und soll nur für Clients aus `IP_NET_1` wirksam werden. `IP_NET_1` ist über `IP_NET_1_DEV` erreichbar. Pakete aus weiteren Netzen sollen nicht berücksichtigt werden.

```
...
  PF_PREROUTING_x='if:ip_net_1_dev:any !@proxy 80 DNAT:@proxy:3128'
      # Anfragen an den http Port, die nicht vom proxy aber über ein
      # internes Interface (ip_net_1_dev) kommen, an den Proxy Port umleiten
      # An dieser Stelle ist es wichtig, mit if:ip_net_1_dev:any zu
      # überprüfen ob die Pakete von innen kommen da sonst auch Pakete von außen
      # umgeleitet werden würden (Sicherheitslücke)

  PF_POSTROUTING_x='prot:tcp IP_NET_1 @proxy:3128 SNAT:IP_NET_1_IPADDR'
      # http-Pakete die aus IP_NET_1 stammen und für den Proxy Port 3128
      # gedacht sind, so umschreiben als wären sie vom fli4l (IP_NET_1_IPADDR)

  PF_FORWARD_x='prot:tcp @proxy 80 ACCEPT'
      # http-Pakete vom proxy durch die Forwardchain durchlassen (wenn nötig)
...
```

3. Beispiel: Um sich das Leben etwas zu erleichtern und die Regeln kürzer zu gestalten, kann man auch Templates einsetzen (vgl. Templates im Paketfilter (Seite 53)). Zweckmäßig ist an dieser Stelle das tmpl:http.

tmpl:http wird zu 'prot:tcp any any:80' und so wird bspw. aus 'tmpl:http IP_NET_1 DNAT:@proxy:3128' 'prot:tcp IP_NET_1 80 DNAT:@proxy:3128'

Sowohl `IP_NET_1` als auch `IP_NET_2` sollen transparent über den Proxy umgeleitet werden. Damit ließe sich vereinfacht auch schreiben:

```
...
  PF_PREROUTING_x='tmpl:http @proxy   ACCEPT'
      # http-Pakete von IP_NET_2 sollen umgeleitet werden

  PF_PREROUTING_x='tmpl:http IP_NET_1 DNAT:@proxy:3128'
      # http-Pakete aus IP_NET_1 sollen umgeleitet werden

  PF_PREROUTING_x='tmpl:http IP_NET_2 DNAT:@proxy:3128'
      # http-Pakete aus IP_NET_2 sollen umgeleitet werden
```

```
  PF_POSTROUTING_x='IP_NET_1 @proxy:3128 SNAT:IP_NET_1_IPADDR'

  PF_FORWARD_x='tmpl:http @proxy ACCEPT'
...
```

Und so ließe sich das endlos fortsetzen ...

3.13.7. DMZ – Demilitarisierte Zone

Fli4l gestattet auch den Aufbau einer einfachen DMZ. Voraussetzung ist ein Router mit drei Interfaces, eines zum WAN (das rote Interface), eines in die DMZ (das orange Interface) und mindestens eins in lokale Netzwerke (grüne Interfaces). Die DMZ kann so konfiguriert werden, dass das im Folgenden beschriebene möglich wird.

Kommunikation zwischen Grün und Rot

Diese Kommunikation wird ganz normal wie oben beschrieben konfiguriert.

Kommunikation zwischen Grün und Orange

Grüne Netze dürfen frei mit Rechnern im orangen Netz kommunizieren. Die Rechner im orangen Netz sehen dabei nichts von der Struktur der grünen Netze; die Verbindungen scheinen alle vom orangen Interface des Routers auszugehen (SNAT).

Die Rechner aus dem orangen Subnetz dürfen keine Verbindungen in grüne Netzwerke hinein aufbauen. Es werden lediglich Pakete durchgelassen, die zu Verbindungen gehören, die von Rechnern eines grünen Netzwerkes aufgebaut wurden.

Kommunikation zwischen Rot und Orange

Verbindungen aus dem roten Netz ins orange Netz können über Portforwarding eingerichtet werden. Die dort aufgeführten Dienste sind aus dem Wan erreichbar. Pakete, die zu solchen Verbindungen gehören, dürfen das orange Netz auch wieder verlassen. Beim Verlassen des orangen Netzes werden die Pakete maskiert, erhalten also die IP des roten Interfaces.

Sollen auch andere Pakete das orange Subnet verlassen, muss das separat konfiguriert werden (DMZ_ORANGE_RED_x).

Kommunikation zwischen Orange und dem Router

Rechner aus dem Orangen Netz können ausgewählte Dienste des Routers nutzen; welche das sind, wird über `DMZ_ORANGE_ROUTER_x` konfiguriert.

Konfiguration der DMZ

Die Typen der Subnetze werden bei ihrer Definition mit festgelegt. Dazu wurde die Variable IP_NET_x_TYPE eingeführt. Die anderen Parameter werden in DMZ_*-Variablen definiert.

OPT_DMZ Aktiviert eine DMZ Konfiguration.

IP_NET_x_TYPE Definiert den Typen des Subnetz. Erlaubte Werte sind 'green' und 'orange'. Wird diese Variable weggelassen, wird das Netz bei der DMZ-Konfiguration ignoriert.

DMZ_RED_DEV Das Interface zum Internet. Dieses kann als `pppoe` für ein dsl-Interface, `ppp<index>` für ein ppp-Device, `ippp<index>` für isdn-Interfaces, als IP_NET_x_DEV für Netzwerk-Interfaces, `eth<index>` für Netzwerkkarten oder `tr<index>` für Token-Ring-Netzwerkkarten angegeben werden.

DMZ_NAT Wenn in der DMZ private Adressen genutzt werden, muss der Router diese maskieren. `DMZ_NAT`='yes' aktiviert das maskieren.

DMZ_ORANGE_RED_N DMZ_ORANGE_RED_x Definiert, welche Verbindungen vom orangen Subnetz aus aufgebaut werden können. Die Syntax entspricht der Syntax des Paketfilters.

DMZ_ORANGE_ROUTER_N DMZ_ORANGE_ROUTER_x Hier wird definiert, welche Dienste des Routers im orangen Subnetz in Anspruch genommen werden können. Die Syntax entspricht der Syntax des Paketfilters.

DMZ_LOG Definiert, ob abgelehnte Pakete vom Kernel protokolliert werden sollen. Dabei können die Meldungen durch Aktivierung von KLOGD an den SYSLOGD über- und durch diesen, entsprechend der Konfiguration ausgegeben werden.

Ein kleines Beispiel mit einem Mail-, Web-Server und einem Web-Proxy in der DMZ könnte wie folgt aussehen:

```
IP_NET_N='3'
IP_NET_1_DEV='eth0'
IP_NET_1='192.168.6.1/24'
IP_NET_1_TYPE='green'
IP_NET_2_DEV='eth1'
IP_NET_2='192.168.7.1/24'
IP_NET_2_TYPE='green'
IP_NET_3_DEV='eth3'
IP_NET_3='192.168.8.1/24'
IP_NET_3_TYPE='orange'

OPT_DMZ='yes'
DMZ_NAT='yes'
DMZ_LOG='yes'
DMZ_RED_DEV='ppp0'
DMZ_ORANGE_RED_N='2'
DMZ_ORANGE_RED_1='prot:tcp 80 ACCEPT' # allow access to http
DMZ_ORANGE_RED_2='prot:tcp 25 ACCEPT' # allow access to smtp
DMZ_ORANGE_ROUTER_N='1'
DMZ_ORANGE_ROUTER_1='53 ACCEPT'       # accept dns requests

PORTFW_N='2'
PORTFW_1_TARGET='80'                  # forward http
PORTFW_1_NEW_TARGET='192.168.8.2'     # ...to dmz host 192.168.8.2
PORTFW_1_PROTOCOL='tcp'               # ...using tcp
PORTFW_2_TARGET='25'                  # forward smtp
PORTFW_2_NEW_TARGET='192.168.8.3'     # ...to dmz host 192.168.8.3
PORTFW_2_PROTOCOL='tcp'               # ...using tcp
```

3.13.8. Port-Forwarding

Für einige Internet-Protokolle ist es nötig, einen Verbindungsaufbau eines Rechners von außen in das interne Netz umzuleiten. Ist das Netz nach außen hin maskiert (IP-Masquerading), also nur eine offizielle IP-Adresse für das gesamte LAN vorhanden, kann man bestimmte Ports oder Protokolle, die von außen erreichbar sein sollen, auf einen bestimmten internen Rechner umleiten. Dieses nennt man Port-Forwarding (bzw. allgemeiner Destination Network Address Translation, kurz DNAT). Die Problematik des Port-Forwarding wird bei der Beschreibung von `MASQ_MODULE_N` noch angesprochen (s.u.).

Oft müssen bei Spielen, Chat oder Internet-Telefonie solche Ports freigeschaltet werden. Können dies die Masquerading-Module selber nicht lösen, kann man selbst Port-Weiterleitungen einrichten.

PORTFW_N Hiermit wird die Anzahl der Port-Weiterleitungen angegeben.
Diese werden in `PORTFW_x_TARGET`, `PORTFW_x_NEW_TARGET` und `PORTFW_x_PROTOCOL` definiert.

PORTFW_x_TARGET Pakete sind an ein bestimmtes Ziel gerichtet (normalerweise eine IP-Adresse und ein Port) und gehören zu einem bestimmten Protokoll. Dieses Ziel ist in einem maskierten Netz der Router, das eigentliche Ziel liegt aber im maskierten Netz, die Pakete müssen also dorthin weitergereicht werden.

`PORTFW_x_TARGET` gibt an, an welches Ziel Pakete gerichtet sein müssen, um weitergeleitet zu werden. Kommt ein solches Paket am Router an, wird es an das eigentliche Ziel weitergeleitet.

Das Ziel wird immer durch eine IP-Adresse, ein Protokoll und evtl. noch einen Port beschrieben. Bei einem Router mit dynamisch zugewiesener IP-Adresse ist die IP zum Zeitpunkt der Konfiguration noch nicht bekannt, kann also nicht angegeben werden. Hier wird das Port-Forwarding erst beim Aufbau der Verbindung eingerichtet, wenn die IP-Adresse bekannt ist und es bezieht sich immer auf Pakete, die am Router ankommen und an die dynamisch zugewiesene IP gerichtet sind.

Im Gegensatz dazu wird das Forwarding bei bekannter IP-Adresse bereits beim Booten eingerichtet. **Wichtig:** *Bei Routern mit fester IP Adresse muss diese hier angegeben werden, sonst wird das Port-Forwarding aufgrund des fehlenden Einwählens nicht korrekt aufgesetzt.*

Folgende Formate sind hier möglich:

`PORTFW_x_TARGET='<port>'`

Leitet alle Pakete weiter, die auf dem Port `<port>` ankommen an einen anderen Rechner weiter.

`PORTFW_x_TARGET='<port1>-<port2>'`

Leitet alle Pakete, die auf den Ports im Bereich zwischen `<port1>` und `<port2>` ankommen, an einen anderen Rechner weiter.

`PORTFW_x_TARGET='<ip>:<port>' PORTFW_x_TARGET='<ip>:<port1>-<port2>'`

Leitet alle Pakete, die auf der IP `<ip>` des Routers auf Port `<port>` (oder im Port-Bereich `<port1>-<port2>`) ankommen, an einen anderen Rechner weiter. Die IP-Adresse kann über einen der in `HOST_x_NAME/HOST_x_ALIAS_x` angegebenen Namen (mit vorangestelltem

'@' – '@hostname) bzw. als Referenz auf IP_NET_x (das in dieser Variable angegebene Netzwerk) oder IP_NET_x_IPADDR (die dort enthaltene IP-Adresse) angegeben werden.

PORTFW_x_TARGET='<circuit>:<port>'

Leitet alle Pakete, die bei eingewähltem Circuit <circuit> an der dazugehörigen IP-Adresse am Port <port> ankommen, an einen anderen Rechner weiter. <circuit> ist entweder die Nummer eines ISDN-Circuits oder "pppoe" für eine DSL-Verbindung (siehe Paket DSL).

PORTFW_x_TARGET='none'

Keine Einschränkung des Ziels. Alle Pakete, die am Router ankommen und zum in PORTFW_x_PROTOCOL angegebenen Protokoll gehören, werden an einen anderen Rechner weitergeleitet.

PORTFW_x_TARGET='<ip>:none'

Alle Pakete, die am Router ankommen, als Zieladresse die angegebene IP-Adresse haben und zum in PORTFW_x_PROTOCOL angegebenen Protokoll gehören, werden an einen anderen Rechner weitergeleitet.

PORTFW_x_NEW_TARGET Dieser Parameter gibt das Ziel eines Port-Forwardings an. Dies ist in der Regel ein anderer Computer im internen Netzwerk. Folgende Formate sind hier möglich:

PORTFW_x_NEW_TARGET='<ip>'

Leitet alle Pakete, die für diese Regel ankommen, an den Rechner mit der IP <ip> weiter. Die IP-Adresse kann über einen der in HOST_x_NAME/HOST_x_ALIAS_x angegebenen Namen (mit vorangestelltem '@' – '@hostname') bzw. als Referenz auf IP_NET_x (das in dieser Variable angegebene Netzwerk) oder IP_NET_x_IPADDR (die dort enthaltene IP-Adresse) angegeben werden.

PORTFW_x_NEW_TARGET='<ip>:<port>'

Leitet alle Pakete, die für diese Regel ankommen, an den Rechner mit der IP <ip> an den Port <port> weiter. Diese Regel ist für Portbereiche (<port1>-<port2>) unzulässig! Die IP-Adresse kann über einen der in HOST_x_NAME/HOST_x_ALIAS_x angegebenen Namen (mit vorangestelltem '@' – '@hostname') bzw. als Referenz auf IP_NET_x (das in dieser Variable angegebene Netzwerk) oder IP_NET_x_IPADDR (die dort enthaltene IP-Adresse) angegeben werden.

PORTFW_x_PROTOCOL Dieser Parameter gibt das Protokoll an, für das diese Regel zutrifft. Zugelassen sind hier 'tcp', 'udp', 'gre' und die Angabe der Protokollnummer (siehe Liste der Protokoll-Nummern der IANA unter http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml.

PORTFW_x_COMMENT Dieser Parameter ist optional. Hier kann man ein Kommentar eingeben. Dieses wird im HTTPD – Status-Webserver (Seite 131) angezeigt.

In früheren Versionen bestand die Möglichkeit, eigene Regeln in einem speziellen Skript abzulegen. Diese Möglichkeit besteht seit fli4l 2.1.0 NICHT mehr, da dieses Skript das variablengesteuerte Port-Forwarding stören würde und eine reibunglose Funktion nicht mehr garantiert werden könnte.

Ich habe auf meiner Homepage ein Skript hinterlegt, welches die Regeln, die man auf dieser Seite findet, in das fli4l-Konfigurationsformat umwandelt. Hiermit können auch Port-Forwarding-Skripte aus alten fli4l-Versionen in das neue Format umgewandelt werden:
`http://portfolio16.de/fli4l/portfw.pl`
Mit dem HTTPD – Status-Webserver (Seite 131) oder dem Windows-Client imonc (Seite 282) ist es möglich, die Regeln zur Laufzeit des Routers zu ändern und die aktualisierte Fassung wieder zurück an den fli4l-Router zu schicken.

3.13.9. Masquerading Module

Die Verwendung von IP-Masquerading hat zwar den Vorteil, dass mehrere Rechner im LAN über eine einzige offizielle IP-Adresse geroutet werden kann, es gibt aber auch Nachteile, die man in Kauf nehmen muss.

Ein großes Problem ist zum Beispiel, dass kein Rechner von außen von sich aus eine Verbindung zu einem Rechner aufnehmen kann. Das ist zwar aus Sicherheitsgründen eigentlich durchaus erwünscht, aber bestimmte Protokolle funktionieren nicht mehr, weil sie einen Verbindungsaufbau von außen einfach erfordern.

Ein klassisches Beispiel ist ftp. Neben dem Kommunikationskanal, auf dem Befehle und Antworten ausgetauscht werden, wird ein weiterer Kanal (in Form eines IP-Ports) verwendet, um die eigentlichen Nutzdaten zu versenden. fli4l verwendet dafür bestimmte Masquerading-Module, um solche zusätzlichen Ports, die verwendet werden, ad hoc dann freizuschalten und an den internen Rechner weiterzuleiten, wenn sie benötigt werden. Dabei "horcht" das Masquerading-Modul in den Datenstrom, um zu erkennen, wann ein zusätzlicher Port benötigt wird.

Typische Anwendungen für Masquerading-Module sind Chat-Protokolle und Spiele im Internet.

MASQ_MODULE_N MASQ_MODULE_x MASQ_MODULE_x_OPTION

Bisher wurde für fli4l das ftp-Masquerading-Modul immer automatisch geladen. Mittlerweile sind weitere Module verfügbar. Diese lassen sich auch mit zusätzlichen Optionen aufrufen. Dies ist zum Beispiel dann nötig, wenn ein FTP-Server auf einem anderen Port als 21 angesprochen werden soll. Hier eine Übersicht aller Module mit ihren Parametern, die vorgegebenen Werte sind die Standardwerte:

ftp File Transfer Protocol

Parameter:

ports=21 Gibt Ports an, die dieses Modul überwachen soll. Wenn also ein FTP-Server auch auf Port 2021 angesprochen werden soll, muss dieser Parameter `ports=21,2021` lauten.

irc Internet Relay Chat

Parameter:

ports=6667 Gibt die Ports an, auf denen ein IRC-Server laufen kann, zu dem man sich verbinden möchte.

Bemerkungen: mIRC ist standardmäßig falsch konfiguriert, so dass er nicht mit diesem Modul zusammenarbeitet. Um mIRC richtig einzustellen, geht man folgendermaßen vor (Verbindung zum IRC-Server erst trennen): Im "Options"-Dialog: Auswahlliste: "Connect", dann "Local Info", hier beide Eingabefelder löschen (falls etwas drin steht), dann "Lookup Method:" auf "Normal" stellen.

pptp PPTP Masquerading

Parameter: keine, horcht immer auf 1723

Bemerkung: Mit diesem Modul lässt sich mehr als ein PPTP-Client gleichzeitig hinter einem Fli4l Router betreiben.

Alle Parameter, die eine Liste von Ports annehmen, können höchstens 8 Ports übergeben bekommen.

Natürlich kann man nicht für jedes Protokoll, welches zusätzliche Ports verwendet, ein separates Masquerading-Modul programmieren. Aus diesem Grund gibt es neben den oben aufgeführten Modulen auch noch eine andere Möglichkeit, neue Ports an einen internen Rechner weiterzuleiten. Siehe dafür auch die Beschreibung zu `PORTFW_N` (Seite 73) (Port-Forwarding). Noch ein Tip: Auf `http://extern.fli4l.de/fli4l_faqengine/faq.php?list=category&catnr=5&prog=1&lang=de&onlynewfaq=1&layout=def` wurde zum Thema Port-Forwarding eine eigene FAQ eingerichtet.

Zu den Konfigurationsvariablen:

`MASQ_MODULE_N` gibt die Anzahl der zu ladenden Masquerading-Module an. `MASQ_MODULE_x` gibt das x'te zu ladende Masquerading-Modul an. Desweitern gibt `MASQ_MODULE_x_OPTION` die Parameter des entspechenden Masquerading- Moduls an. Hier können die oben beschriebenen Parameter, mit Leerzeichen getrennt angegeben werden.

Die Standardeinstellung für `MASQ_MODULE_N` ist '1'. Damit wird nur das erste Modul, nämlich ftp, geladen. Möchte man weitere Module laden, ist `MASQ_MODULE_N` entsprechend zu erhöhen. Die Reihenfolge in der beigefügten Datei config/base.txt ist nur ein Beispiel. Sie kann also beliebig geändert werden, z.B.

```
MASQ_MODULE_N='2'
MASQ_MODULE_1='ftp'
MASQ_MODULE_1_OPTION='ports=21,2121'
MASQ_MODULE_2='irc'
MASQ_MODULE_2_OPTION=''
```

Weitere Inhalte von `MASQ_MODULE_x` werden einfach ignoriert.

Hier wurde auch gleich ein Beispiel für die Parameter des FTP-Moduls angegeben.

3.14. Domain-Konfiguration

Windows-PCs im LAN haben unter anderem ;-) eine unangenehme Eigenschaft: Sobald ein Nameserver benötigt wird und man diesen deshalb im Windows einstellt, fragen diese Windows-Rechner den angegebenen Nameserver in regelmäßigen Abständen ab (bei mir so alle 5 Minuten) – auch wenn man gar nicht daran arbeitet!

Würde man also auf dem Windows-PC einen DNS-Server im Internet angeben, wird's teuer :-(

Der Trick ist nun folgender: Wenn im LAN nicht bereits ein DNS-Server vorhanden ist, kann man das DNS im fli4l-Router verwenden.

Es wird DNSMASQ als DNS-Server eingesetzt.

Wenn man jedoch mit der DNS-Konfiguration beginnt, sollte man sich zunächst Gedanken über den Domain-Namen und die Namen der PCs im Netz machen. Der verwendete Domain-Name wird nicht im Internet sichtbar. Deshalb kann man sich hier prinzipiell beliebige Domain-Namen ausdenken.

Ausserdem sollte man jedem Windows-Rechner einen Namen verpassen. Diese Namen müssen dem fli4l-Router bekannt sein.

DOMAIN_NAME In der Auslieferung ist als Domain einfach "lan.fli4l" eingestellt. Hier kann sich jeder austoben, da die lokal verwendete Domain nicht im Internet sichtbar wird. Sie sollten lediglich vermeiden, einen Namen zu benutzen, den es im Internet geben könnte (z.B. irgendwas.de), da Sie sonst nicht auf diese Domain werden zugreifen können.

DNS_FORWARDERS Hier ist die DNS-Server-Adresse des Internet-Providers anzugeben, wenn fli4l als Router in das Internet verwendet wird. Der fli4l-Router gibt dann sämtliche DNS-Queries, die er nicht selbst beantworten kann, an diese Adresse weiter.

Möchte man mehrere DNS-Forwarder angeben, trennt man die IP-Adressen durch Blanks.

Es ist auch möglich, optional Port-Nummern an die IP-Adressen durch Doppelpunkt getrennt anzugeben. Allerdings muss dann `OPT_DNS='yes'` (Seite 103) sein (Paket dns_dhcp (Seite 101)) und es darf nirgends die Option `*_USEPEERDNS` benutzt werden.

Achtung: Auch wenn

- `PPPOE_USEPEERDNS` (Seite 110),
- `ISDN_CIRC_x_USEPEERDNS` (Seite 151) oder
- `DHCP_CLIENT_x_USEPEERDNS` (Seite 101)

gesetzt (='yes') ist, ist hier die Eintragung eines Servers nötig, da sonst direkt nach dem Start keine Namensauflösung möglich ist.

Ausnahme: fli4l als Router in einem lokalen Netz *ohne* Anschluss an das Internet oder (Firmen-)Netze mit weiteren DNS-Servern. In diesem Fall ist 127.0.0.1 anzugeben, um das Forwarding zu unterbinden.

HOSTNAME_IP (optional)

Hiermit kann optional festgelegt werden, an welches Netz 'IP_NET_x' der `HOSTNAME` gebunden wird.

HOSTNAME_ALIAS_N (optional)

Anzahl der zusätzlichen Alias-Hostnamen für den Router.

HOSTNAME_ALIAS_x (optional)

Zusätzlicher Alias-Name für den Router.

3.15. imond-Konfiguration

START_IMOND Mit `START_IMOND` kann man einstellen, ob der imond-Server aktiviert werden soll. imond übernimmt dabei das Monitoring/Controlling und Least-Cost-Routing des fli4l-Routers. Der Beschreibung von imond (Seite 277) ist deshalb ein extra Kapitel gewidmet (s.u.).

Wichtig: Die LC-Routing-Features von fli4l können nur mit imond genutzt werden. Ein zeitabhängiges Umschalten von Verbindungen ist ohne imond nicht möglich!

Für ISDN- und DSL-Routing ist imond ab Version 1.5 zwingend erforderlich. In diesem Fall ist `START_IMOND`='yes' einzustellen.

Wird fli4l lediglich als Router zwischen 2 Netzwerken eingesetzt, sollte `START_IMOND`='no' eingestellt werden.

Standard-Einstellung: `START_IMOND`='no'

IMOND_USE_ORIG Wählt zwischen normalem und experimentellem imond. Nur für Entwickler gedacht, sollte auf 'yes' gelassen werden.

IMOND_PORT TCP/IP-Port, auf dem imond auf Verbindungen horcht. Der Standard-Wert '5000' sollte nur in Ausnahmefällen geändert werden.

IMOND_PASS Hier kann ein spezielles User-Password für imond gesetzt werden. Meldet sich ein Client auf Port 5000 an, erwartet imond (und damit auch seine Clients) die Eingabe dieses Passworts, bevor er irgendeinen Befehl korrekt beantwortet. Ausnahme: Befehle "quit", "help" und "pass". Ist `IMOND_PASS` leer, wird kein Password benötigt.

Ob der Client im User-Modus bestimmte Steuerbefehle, wie Dial, Hangup, Reboot, Umschalten der Default-Route bereits ausführen kann oder dafür die Eingabe des Admin-Passwords zwingend notwendig ist, wird über die Variablen

- IMOND_ENABLE (Seite 80),
- IMOND_DIAL (Seite 80),
- IMOND_ROUTE (Seite 80) und
- IMOND_REBOOT (Seite 80)

eingestellt, siehe unten.

Standard-Einstellung: `IMOND_PASS`="

IMOND_ADMIN_PASS Mit Hilfe der Admin-Passwords erhält der Client alle Rechte und kann so sämtliche Steuerfunktionen des imond-Servers nutzen – und zwar unabhängig von den Variablen `IMOND_ENABLE`, `IMOND_DIAL` usw. Lässt man `IMOND_ADMIN_PASS` leer, so reicht die Eingabe des User-Passwords, um sämtliche Rechte zu erhalten!

Standard-Einstellung: `IMOND_ADMIN_PASS`="

IMOND_LED imond kann den Online/Offline-Status nun über eine LED anzeigen. Diese wird folgendermaßen an einen COM-Port angeschlossen:

Verbindung 25-polig:

```
20 DTR  -------- 1kOhm ----- >| ---------- 7 GND
```

Verbindung 9-polig:

```
4 DTR  -------- 1kOhm ----- >| ---------- 5 GND
```

Ist eine ISDN- oder DSL-Verbindung aufgebaut, leuchtet die LED. Ansonsten ist sie ausgeschaltet. Sollte es genau umgekehrt sein, ist die Leuchtdiode umzupolen. Sollte die LED zu schwach leuchten, kann der Vorwiderstand bis auf 470 Ohm reduziert werden.

Es ist auch möglich, zwei verschiedenfarbige LEDs anzuschließen. Dann ist die zweite LED ebenso über einen Vorwiderstand zwischen DTR und GND anzuschließen, jedoch genau umgekehrt. Dann leuchtet je nach Zustand die eine oder die andere LED. Oder man verwendet direkt eine DUO-LED (zweifarbig, drei Anschlussbeinchen).

Im Moment verhält sich der RTS-Anschluss der seriellen Schnittstelle genauso wie DTR. Hier könnte also noch eine weitere LED angechlossen werden, die den Online/Offline-Zustand anzeigt. Das könnte sich jedoch in einer zukünftigen fli4l-Version ändern.

Als Wert von `IMOND_LED` muss ein COM-Port angegeben werden, also 'com1', 'com2', 'com3' oder 'com4'. Ist keine LED angeschlossen, sollte die Variable leer gelassen werden.

IMOND_BEEP Mit `IMOND_BEEP`='yes' gibt imond einen Zweiklang-Ton über den PC-Lautsprecher aus, wenn der Zustand von Offline nach Online wechselt und umgekehrt. Im ersten Fall wird zuerst ein tiefer, dann ein hoher Ton ausgegeben. Beim Wechsel in den Offline-Status zurück wird zuerst der höhere, dann der tiefere Ton ausgegeben.

IMOND_LOG Wird `IMOND_LOG`='yes' benutzt, werden in der Datei `/var/log/imond.log` die Verbindungen protokolliert. Diese Datei kann z.B. für Statistikzwecke per scp auf einen Rechner im LAN kopiert werden. Für den scp-Zugriff ist aber dann noch das Paket sshd zu installieren und so zu konfigurieren, dass es auch scp zur Verfügung stellt.

Das Format der Logdateieinträge ist in Tabelle 3.7 beschrieben.

Tabelle 3.7.: Format der Imond-Logdatei

Eintrag	Bedeutung
Circuit	der Name des Circuits, für den der Eintrag erzeugt wurde
Startzeit	Datum und Uhrzeit der Einwahl dieses Circuits
Stopzeit	Datum und Uhrzeit des Auflegens dieses Circuits
Online-Zeit	die Zeit, die dieser Circuit online war
Abgerechnete Zeit	die Zeit, die der Provider abrechnen wird (hängt vom Takt ab)
Kosten	die Kosten, die der Provider für die Zeit in Rechnung stellt
Bandbreite	die genutzte Bandbreite getrennt nach in und out (in zuerst), dargestellt als zwei vorzeichenlose Integerzahlen, für die gilt: Bandbreite = $4GiB * <erste\ Zahl> + <zweite\ Zahl>$
Device	das Gerät, über das kommuniziert wurde
Abrechnungstakt	der Takt, das vom Provider zur Abrechnung herangezogen wird (Daten der Circuit-Konfiguration)
Taktgebühren	die Gebühren, die pro Takt fällig werden (Daten der Circuit-Konfiguration)

Die Kosten werden in Euro ausgegeben. Wichtig ist dabei die korrekte Definition der entsprechenden Circuit-Variablen `ISDN_CIRC_x_TIMES` (Seite 158).

Standard-Einstellung: `IMOND_LOG`='no'

IMOND_LOGDIR Ist das Protokollieren eingeschaltet, kann über `IMOND_LOGDIR` ein alternatives Verzeichnis statt /var/log angegeben werden, z.B. '/boot'. Dann wird die Log-Datei imond.log auf der Diskette angelegt. Dazu muss dann die Diskette auch Read/Write "gemounted" sein.

IMOND_ENABLE IMOND_DIAL IMOND_ROUTE IMOND_REBOOT Durch diese Variablen werden bestimmte Kommandos, die von imonc-Clients zum imond-Server gesendet werden, bereits im User-Modus freigeschaltet.

Hiermit kann man einstellen, ob der imond-Server die ISDN-Schnittstelle ein- bzw. ausschalten, wählen/einhängen, eine neue Default-Route setzen und/oder den Rechner booten darf.

Standard-Einstellungen:

```
IMOND_ENABLE='yes'
IMOND_DIAL='yes'
IMOND_ROUTE='yes'
IMOND_REBOOT='yes'
```

Alle weiteren Features der Client-/Server-Schnittstelle von imond sind in einem eigenen Kapitel (Seite 277) beschrieben.

3.16. Allgemeine Circuit-Konfiguration

IP_DYN_ADDR Wird eine Verbindung mit dynamischer Adressvergabe verwendet, ist `IP_DYN_ADDR` auf 'yes' zu stellen, ansonsten auf 'no'. Die meisten Internet-Provider verwenden eine dynamische Adressvergabe.

Standard-Einstellung: `IP_DYN_ADDR`='yes'

DIALMODE Der Standard-Wählmodus von fli4l ist 'auto', d.h. es wird automatisch gewählt, wenn eine IP-Adresse außerhalb des eigenen Netzes angesprochen wird. Es ist aber auch möglich, als Dialmode 'manual' oder 'off' anzugeben. In diesem Fall kann man eine Wählverbindung nur über den Client imonc auslösen.

Standard-Einstellung: `DIALMODE`='auto'

4. Pakete

Neben der Basisinstallation (BASE) gibt es Pakete. Diese enthalten ein oder mehrere "OPTs"[1], die bei Bedarf zu BASE hinzuinstalliert werden können. Einige dieser Pakete sind Bestandteil des Basis-Paketes, andere kommen separat. Eine Übersicht über die vom Fli4l-Team bereitgestellten Pakete finden Sie auf der Download-Seite (http://www.fli4l.de/download/stabile-version/), die von anderen Autoren bereitgestellten Pakete sind in der OPT-Datenbank (http://extern.fli4l.de/fli4l_opt-db3/) zu finden. Im folgenden werden die vom Fli4l-Team bereitgestellten Pakete beschrieben.

Nicht alle Pakete wird man zur gleichen Zeit auf die Diskette bekommen. Bei einer minimalen fli4l-Konfiguration,bestehend aus den Paketen BASE, KERNEL 2.6.16 und PPPOE, bleiben ca. 400 kB komprimierter Speicherplatz auf der Diskette frei. Diese können dann für weitere Pakete eingesetzt werden.

4.1. Werkzeuge im Basispaket

Im Basispaket befinden sich die folgenden OPTs:

Name	Beschreibung
OPT_SYSLOGD	Werkzeug zum Protokollieren von Systemmeldungen (Seite 81)
OPT_KLOGD	Werkzeug zum Protokollieren von Kernelmeldungen (Seite 83)
OPT_LOGIP	Werkzeug zum Protokollieren von WAN-IP-Adressen (Seite 83)
OPT_Y2K	Datumskorrektur bei nicht Y2K-festen Rechnern (Seite 83)
OPT_PNP	Installation der isapnp-Werkzeuge (Seite 84)

4.1.1. OPT_SYSLOGD – Protokollieren von Systemmeldungen

Viele Programme verwenden die Syslog-Schnittstelle, um Meldungen auszugeben. Damit diese auch auf der Konsole sichtbar werden, muss in diesem Falle der Daemon syslogd gestartet werden.

Sind Debug-Meldungen gewünscht, stellt man OPT_SYSLOGD auf 'yes', ansonsten auf 'no'.

Siehe auch ISDN_CIRC_x_DEBUG (Seite 157) und PPPOE_DEBUG (Seite 111).

Standard-Einstellung: OPT_SYSLOGD='no'

SYSLOGD_RECEIVER Mit SYSLOGD_RECEIVER kann man festlegen, ob fli4l Syslog-Nachrichten vom Netzwerk empfangen soll oder nicht.

SYSLOGD_DEST_N SYSLOGD_DEST_x Mit SYSLOGD_DEST_x gibt man Ziele an, wohin die System-Meldungen, die von syslogd entgegengenommen werden, ausgegeben werden. Im Normalfall ist dies die Konsole von fli4l, also

```
SYSLOGD_DEST_1='*.* /dev/console'
```

[1] Abk. für "OPTionales Modul"

Möchte man eine Datei als Protokoll verwenden, ist z.B. einzutragen:

```
SYSLOGD_DEST_1='*.* /var/log/messages'
```

Der Syslog-Daemon setzt in regelmässigen Zeitabstand einen — MARK — Eintrag falls es nichts zu protokollieren gibt. Damit soll ein eventueller Ausfall des Syslogd im Log nachvollziehbar sein. Allerdings kann man darauf verzichten wenn das System stabil läuft oder das Logfile übersichtlich bleiben soll.

```
SYSLOGD_DEST_1='*.*;mark.!info /var/log/messages'
```

Ist ein sog. Loghost im Netz vorhanden, können die Meldungen auch auf diesen Rechner umgeleitet werden – unter Angabe der IP-Adresse.

Beispiel:

```
SYSLOGD_DEST_1='*.* @192.168.4.1'
```

Das @-Zeichen ist dann der IP-Adresse voranzustellen.

Wenn die Systemmeldungen an mehrere Ziele "ausgeliefert" werden sollen, ist es nötig, die Variable `SYSLOGD_DEST_N` (Anzahl der Ziele) entsprechend zu erhöhen und die Variablen `SYSLOG_DEST_1`, `SYSLOG_DEST_2` usw. zu füllen.

Der Parameter '*.*' bedeutet, dass sämtliche Meldungen protokolliert werden. Man kann jedoch die Meldungen für bestimmte Ziele über sog. "Prioritäten" einschränken. In diesem Fall ersetzt man das Sternchen (*) hinter dem Punkt (.) durch eines der folgenden Schlüsselwörter:

- debug
- info
- notice
- warning (veraltet: warn)
- err (veraltet: error)
- crit
- alert
- emerg (veraltet: panic)

Die Reihenfolge in der Liste spiegelt dabei das "Gewicht" der Meldungen wider. Die Schlüsselwörter "error", "warn" und "panic" sind veraltet und sollten nicht mehr verwendet werden.

Vor dem Punkt kann eine sog. "Facility" statt des Sternchens (*) eingetragen werden. Eine Erklärung würde aber hier zu weit gehen. Der geneigte Leser kann hierfür eine Suchmaschine seiner Wahl bemühen. Eine Übesicht über mögliche Facilities finden sich auf den Manpages der syslog.conf:

http://linux.die.net/man/5/syslog.conf

Normalerweise ist das Sternchen aber vollkommen ausreichend. Beispiel:

```
SYSLOGD_DEST_1='*.warning @192.168.4.1'
```

Nicht nur Unix-/Linux-Rechner können als Loghost dienen, sondern auch Windows-Rechner. Auf http://www.fli4l.de/sonstiges/links/ findet man Verweise auf entsprechende Software. Die Verwendung eines Loghosts wird dringend empfohlen, wenn eine detaillierte Protokollierung gewünscht ist. Auch hilft die Protokollierung bei der Fehlersuche. Mittlerweile "versteht" auch imonc als Windows-Client das Syslog-Protokoll und kann die Meldungen in einem Fenster ausgeben.

Leider lassen sich die Boot-Meldungen von fli4l nicht über syslogd umlenken. Jedoch kann man fli4l auch so konfigurieren, dass die Konsole ein serielles Terminal (bzw. Terminalemulation) ist. Wie das geht, steht in dem Abschnitt Konsolen-Einstellungen (Seite 30).

SYSLOGD_ROTATE Mit SYSLOGD_ROTATE kann man festlegen, ob fli4l Syslog-Nachrichten einmal täglich rotiert. Es werden dabei die Meldungen der letzten Tage archiviert.

SYSLOGD_ROTATE_DIR Mir der optionalen Variable SYSLOGD_ROTATE_DIR, kann man festlegen, dass die rotierten Syslog-Dateien nicht in /var/log/ sondern in das angegebene Directory rotiert werden.

4.1.2. OPT_KLOGD – Protokollieren von Kernelmeldungen

Viele der auftretenden Fehler – z.B. fehlgeschlagene Einwahl – werden vom Linux-Kernel direkt auf die Konsole geschrieben. Mit OPT_KLOGD='yes' werden diese Meldungen an den syslogd umgelenkt, welcher diese dann entweder in Dateien protokollieren oder an einen Loghost weiterleiten kann, s.o. Dann hat man auf der fli4l-Konsole (fast) Ruhe.
Empfehlung: Wenn Sie OPT_SYSLOGD='yes' benutzen, sollte man auch OPT_KLOGD='yes' setzen.
Standard-Einstellung: OPT_KLOGD='no'

4.1.3. OPT_LOGIP – Protokollieren von WAN-IP-Adressen

Mit LOGIP ist es möglich, die WAN-IP in einer Log-Datei festzuhalten. Mit Angabe von OPT_LOGIP='yes' wird die Funktion aktiviert.
Standard-Einstellung: OPT_LOGIP='no'

LOGIP_LOGDIR Verzeichnis der Log-Datei festlegen

Mit LOGIP_LOGDIR wird das Verzeichnis festgelegt, in welchem die Log-Datei angelegt wird.

Empfehlung: LOGIP_LOGDIR sollte auf '/boot' bei einer Disketteninstallation gesetzt werden, auf '/data' bei eine HD-Install mit Daten-Partition.

Standard-Einstellung: LOGIP_LOGDIR='/boot'

4.1.4. OPT_Y2K – Datumskorrektur bei nicht Y2K-festen Rechnern

Meist werden fli4l-Router aus alten Hardware-Teilen zusammengesetzt. Dabei kann das BIOS des Mainboards nicht Y2K-fest sein. Das kann dazu führen, dass bei einer Einstellung des Datums auf den 27.05.2000 im BIOS beim nächsten Booten der 27.05.2094 im BIOS zu finden ist! Linux zeigt dann übrigens den 27.05.1994 an.

Eigentlich kann das eingestellte Datum für den fli4l-Router egal sein und sollte deshalb keine Rolle spielen. Wird fli4l jedoch als Least-Cost-Router eingesetzt, kann dies sehr wohl eine Rolle spielen.

Grund: Der 27.05.1994 war ein Freitag, der 27.05.2000 jedoch ein Samstag. Und am Wochenende gibt's günstigere Tarife bzw. günstigere Provider ...

Eine erste Lösung lautet: Das BIOS-Datum wird vom 27.05.2000 auf den 28.05.1994 gestellt. Das war ebenso ein Samstag. Damit ist das Problem aber noch nicht gänzlich gelöst, denn fli4l verwendet nicht nur den Wochentag und die momentane Uhrzeit für das LC-Routing, sondern berücksichtigt auch die Feiertage.

Y2K_DAYS – N Tage auf Systemdatum addieren

Da das gesetzte Datum genau 2191 Tage zum tatsächlichen Datum differiert, werden bei Angabe von

```
Y2K_DAYS='2191'
```

auf das BIOS-Datum 2191 Tage addiert und dieses dann als Linux-Datum gesetzt. Das BIOS-Datum bleibt davon jedoch unberührt, sonst würde beim nächsten Booten das Datum wieder das Jahr 2094 (bzw. 1994) aufweisen.

Es gibt noch eine Alternative:

Mit dem Zugriff auf einen Time-Server kann sich fli4l die aktuelle Datum/Uhrzeit aus dem Internet holen. Dafür steht das Paket CHRONY (Seite 98) zur Verfügung. Beide Einstellungen lassen sich kombinieren. Das ist sinnvoll, um mit Y2K_DAYS zunächst das Datum schon einmal zu korrigieren und anschließend über den Time-Server die genaue Uhrzeit einzustellen.

Wer mit Y2K keine Probleme hat: Variable OPT_Y2K='no' setzen und einfach nicht mehr darüber nachdenken ...

4.1.5. OPT_PNP – Installation von isapnp tools

Teilweise müssen ISAPnP-Karten über das Werkzeug "isapnp" konfiguriert werden. Dies betrifft insb. ISDN-Karten mit ISDN_TYPE 7, 12, 19, 24, 27, 28, 30 und 106 – aber nur, wenn es sich auch wirklich um ISAPnP-Karten handelt.

Zur Konfiguration ist die Erstellung einer Konfigurations-Datei etc/isapnp.conf notwendig.

Hier eine Kurzanleitung zur Erstellung:

- In <config>/base.txt die Variable OPT_PNP='yes' und MOUNT_BOOT='rw' setzen
- fli4l booten – die ISAPnP-Karte wird wahrscheinlich nicht erkannt
- Auf fli4l-Konsole eingeben:

```
pnpdump -c >/boot/isapnp.conf
umount /boot
```

Damit ist die Konfiguration auf der Diskette gespeichert.

Weiter auf PC (Unix/Linux/Windows):

- Die Datei isapnp.conf von Diskette nach <config>/etc/isapnp.conf kopieren
- isapnp.conf mit Editor öffnen, bearbeiten und abspeichern
  Die vorgegebenen Werten kann man hier beibehalten oder auch ersetzen von anderen, die man aus den möglichen Werten wählt. Relevant sind dabei die folgenden Zeilen im folgenden Beispiel:

```
      #     Start dependent functions: priority acceptable
      #       Logical device decodes 16 bit IO address lines
      #             Minimum IO base address 0x0160
      #             Maximum IO base address 0x0360
      #             IO base alignment 8 bytes
      #             Number of IO addresses required: 8
1)       (IO 0 (SIZE 8) (BASE 0x0160))
      #       IRQ 3, 4, 5, 7, 10, 11, 12 or 15.
      #             High true, edge sensitive interrupt (by default)
2)       (INT 0 (IRQ 10 (MODE +E
```

  1) – hier kann als „BASE“ eine Adresse zwischen die angegebene Minimum und Maximum eingegeben werden, wobei man das „base alignment“ in betracht ziehen muss. Bei mehr als einer isa-Karte im System muss immer darauf geachtet werden, dass es hier keine Überschneidungen gibt, achte dabei auch auf die benötigte Anzahl Adressen (number of addresses required).

  2) – Hier kann aus der angezeigten Liste ein IRQ eingesetzt werden. Dabei ist 2(9), 3, 4, 5 und 7 eher eine schlechte Wahl, da sich diese normalerweise mit den Seriellen und Parallelen Schnittstellen bzw der Cascadierung ins Gehege kommen.
  ISA Karten können IRQs nicht teilen, deshalb darf ein für diese Karte verwendeter nicht anderweitig belegt sein.

- Die entsprechenden Daten (IRQ/IO) in die <config>/isdn.txt übernehmen
- Es ist nötig in der <config>/base.txt die `OPT_PNP` Einstellung auf ‘yes’ zu belassen, anderenfalls werden die erforderliche Dateien nicht mit auf die Diskette kopiert. Die Einstellung `MOUNT_BOOT` kann man beliebig ändern.
- Neue Boot-Diskette erzeugen

Die automatisch generierte Datei ist im Unix-Format gespeichert und enthält keine CRs. Startet man unter Windows den Notepad-Editor, zeigt dieser alle Zeilen in einer einzigen Zeile an. Der DOS-Editor “edit” kann jedoch mit Unix-Dateien umgehen. Er speichert sie dann als DOS-Datei (mit CRs) ab.

Abhilfe:

- DOS-Box starten
- In das Verzeichnis <config>/etc wechseln
- Eingeben: edit isapnp.conf
- Datei bearbeiten und abspeichern

Anschließend kann die Datei auch wieder mit Notepad bearbeitet werden.

Man kann auch unter Windows einfach den Wordpad-Editor verwenden.

Die zusätzlich generierten CRs werden beim Booten von fli4l wieder herausgefiltert. Sie stören also nicht.

Zunächst sollte man versuchen, ohne `OPT_PNP` auszukommen. Wird die Karte nicht erkannt, sollte wie oben beschrieben vorgegangen werden.

Bei einem Update auf eine neuere fli4l-Version kann die früher erstellte Datei isapnp.conf weiter verwendet werden.

Standard-Einstellung: `OPT_PNP='no'`

4.2. Advanced Networking

Das Advanced Networking Paket beinhaltet die Möglichkeit den fli4l–Router um VLAN, Bonding und Bridging Funktionen zu erweitern. Zusätzlich gibt es die Möglichkeit EBTables (http://ebtables.sourceforge.net/) Unterstützung zu aktivieren. Damit wird es möglich einen transparenten Paketfilter aufzubauen.

Generell gilt bei allen Paketen aus dem advanced_networking Paket:

Dieses Paket ist nur für Anwender gedacht, die sich sehr gut im Bereich Netzwerke auskennen. Insbesondere sind fundierte Routingkenntnisse notwendig.

Gerade beim aktivieren der EBTables Unterstützung können sehr ungewöhnliche Probleme auftreten, wenn man sich nicht 100% mit den verschiedenen Wirkungsweisen von Layer 2 und 3 auskennt. Einige Paketfilterregeln arbeiten mit aktivierter EBTables Unterstützung vollkommen anders als gewohnt.

4.2.1. Bonding - mehrere Netzwerkkarten zusammenfassen zu einem Link

Unter Bonding versteht man das Zusammenfassen von mindestens zwei Netzwerkkarten, die auch unterschiedlichen Typs (also 3Com und Intel) und Geschwindigkeit (10 Mbit/s oder 100 Mbit/s) sein können, zu einer gemeinsamen Verbindung. Dabei können entweder entsprechende Linux Rechner direkt verbunden werden, oder eine Verbindung zu einem Switch aufgebaut werden. So kann z.B. ohne grossen Aufwand eine 200 Mbit/s Full-Duplex Verbindung vom fli4l-Router zu einem Switch geschaltet werden. Jeder der sich für Bonding interessiert sollte vorher die Dokumentation dazu im Kernelverzeichnis (bonding.txt) gelesen haben. Die Namen der Bondingeinstellungen entsprechen weitestgehend den dort verwendeten Namen. Unter dem Linuxkernel 2.6.x findet sich im Verzeichnis der Kernelsourcen unter Documentation/networking die Datei bonding.txt.

OPT_BONDING_DEV Default: `OPT_BONDING_DEV='no'`

Mit `'yes'` wird das Bonding Paket aktiviert. Die Einstellung `'no'` deaktiviert das Bonding Paket komplett.

BONDING_DEV_N Default: `BONDING_DEV_N='0'`

Die Anzahl der zu konfigurierenden Bondinggeräte.

BONDING_DEV_x_DEVNAME Default: `BONDING_DEV_x_DEVNAME="`

Der Name des Bondinggerätes das erstellt werden soll. Der Name muß dabei mit 'bond' beginnen und es muß eine Zahl ohne führende '0' folgen. Die Namen der Bondinggeräte müssen nicht mit '0' beginnen und müssen nicht aufeinanderfolgend sein. Mögliche Werte wären z.B. 'bond0', 'bond8' oder 'bond99'.

BONDING_DEV_x_MODE Default: `BONDING_DEV_x_MODE="`

Gibt eines der Bonding-Methoden an. Der Standardwert ist Round-Robin 'balance-rr'. Mögliche Werte sind hier aufgelistet:

balance-rr Round-Robin-Methode: Übermittle der Reihe nach über alle Slaves von ersten bis zum letzten. Diese Methode bietet sowohl Load- Balancing als auch Fehlertoleranz.

active-backup Aktives Backup: Nur ein Slave im Bond ist aktiv. Die anderen Slaves werden nur dann aktiviert, wenn der aktive Slave ausfällt. Die MAC-Adresse des Bonds ist nur auf einem Port (Netzwerkadapter) sichtbar, um den Switch nicht zu verwirren. Dieser Modus bietet Fehlertoleranz.

balance-xor XOR-Methode: Übermittle basierend auf der Formel [(Quell-MAC-Adresse XOR Ziel-MAC-Adresse) modulo Anzahl der Slaves]. Dadurch wird immer der selbe Slave für die selben Ziel-MAC-Adresse benutzt. Diese Methode bietet sowohl Load-Balancing als auch Fehlertoleranz.

broadcast Broadcast-Methode: Übermittelt alles auf allen Slave-Devices. Dieser Modus bietet Fehlertoleranz.

802.3ad Dynamische IEEE 802.3ad Verbindungsaggregation. Erstellt Aggregationsgruppen, die die selben Geschwindigkeits und Duplex- Einstellungen teilen. Übermittelt auf allen Slaves im aktiven Aggregator.

Voraussetzungen:

- Unterstützt für ethtool im Basistreiber,um Geschwindigkeit und Duplex-Status für jede Device abzufragen.
- Ein Switch, der dynamische IEEE 802.3ad Verbindungsaggregation unterstützt.

balance-tlb Adaptives Load-Balancing für ausgehende Daten: Kanal-Bonding, dass keine speziellen Features im Switch benötigt. Der ausgehende Netzwerktraffic wird entsprechend der momentanen Last (relativ zur Geschwindigkeit berechnet) auf jeden Slave verteilt. Eingehender Netzwerktraffic wird vom aktuellen Slave empfangen. Wenn der empfangende Slave ausfällt, übernimmt ein anderer Slave die MAC-Adresse des ausgefallenen Empfangsslaves.

Voraussetzungen:

Unterstützt für ethtool im Basistreiber,um Geschwindigkeit und Duplex-Status für jede Device abzufragen.

balance-alb Adaptives Load-Balancing: schliesst sowohl balance-tlb, als auch Eingehendes Load-Balancing (rlb) für IPV4 Traffic ein und benötigt keine speziellen Voraussetzungen beim Switch. Load- Balancing für eingehenden Traffic wird über ARP-Absprache erreicht. Der Bonding-Treiber fängt ARP-Antworten vom Server auf ihrem Weg nach aussen hin ab und überschreibt die Quell- Hardware-Adresse mit

der eindeutigen HW-Adresse eines Slaves im Bond, so dass unterschiedliche Clients unterschiedliche HW-Adressen für den Server verwenden.

Eingehender Traffic von Verbindungen, die vom Server erstellt wurden wird auch verteilt. Wenn der Server ARP-Anfragen sendet, kopiert und speichert der Bonding-Treiber die Client-IP aus dem ARP. Wenn die ARP-Antwort des Client ankommt, wird seine HW-Adresse ermittelt und der Bonding-Treiber erstellt eine ARP-Antwort an diesen Client und ordnet ihn so zu einem Client im Bond zu. Ein Problematischer Effekt von ARP-Absprachen für die Lastverteilung ist, dass jedes Mal wenn eine ARP-Anfrage übermittelt wird, sie die HW-Adresse des Bonds benutzt. Also lernen die Clients die HW-Adresse des Bonds und der eingehende Traffic auf dem aktuellen Slave bricht zusammen. Diesem Umstand wird begegnet, indem Updates (ARP-Antworten) zu allen Clients mit ihrer jeweiligen HW-Adresse gesandt wird, sodass der Traffic wieder aufgeteilt ist. Eingehender Traffic wird auch dann neu aufgeteilt, wenn ein neuer Slave zum Bond hinzugefügt wird oder ein inaktiver Slave reaktiviert wird. Die Empfangslast wird der Reihe nach (Round-Robin) in der Gruppe der Slave mit der grössten Geschwindigkeit im Bond verteilt.

Wenn eine Verbindung wiederhergestellt wird oder ein neuer Slave zum Bond hinzukommt wird der eingehende Traffic neu auf alle aktiven Slaves im Bond verteilt, indem ARP-Antworten mit den ausgewählten MAC-Adressen zu jedem Client gesandt werden. Der Parameter updelay muss auf einen Wert grösser oder gleich der Weiterleitungsverzögerung (forwarding delay) des Switchs eingestellt sein, sodass ARP-Antworten an die Clients nicht vom Switch geblockt werden.

Voraussetzungen:

- Unterstützt für ethtool im Basistreiber,um Geschwindigkeit und Duplex-Status für jede Device abzufragen.
- Unterstützung im Basistreiber, die HW-Adresse auch dann setzen zu können, wenn das Device offen ist. Das ist notwendig, damit immer ein Slave im Team die HW-Adresse des Bonds tragen kann, (der curr_active_slave) obwohl jeder Slave im Bond eine eigene, eindeutige HW-Adresse hat. Wenn der curr_active_slave ausfällt, wird seine HW-Adresse mit dem neuen curr_active_slave ausgetauscht.

BONDING_DEV_x_DEV_N Default: `BONDING_DEV_x_DEV_N='0'`

Gibt an, aus wievielen Geräten dieses Bondinggeräte besteht. Wenn z.B. ein Bondinggerät aus 'eth0' und 'eth1' gebildet werden soll muß hier eine '2' (für die beiden eth–Geräte) eingetragen werden.

BONDING_DEV_x_DEV_x Default: `BONDING_DEV_x_DEV_x="`

Der Name eines Gerätes, welches zu diesem Bondinggerät gehören soll. Ein möglicher Wert wäre z.B. 'eth0'. Bitte beachten Sie, dass ein Gerät, welches Sie für ein Bondinggerät benutzen, exklusiv dafür benutzt werden muß. Insbesondere ist es nicht möglich das Gerät für ein DSL-Modem, eine Bridge, ein VLAN oder in der base.txt zu benutzen.

BONDING_DEV_x_MAC Default: `BONDING_DEV_x_MAC="`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Ein Bondinggerät benutzt standardmäßig die MAC Adresse des ersten Gerätes, welches für das Bonding benutzt wird. Wenn Sie dies nicht wollen können Sie auch eine MAC Adresse angegeben, die das Bondinggerät benutzen soll.

BONDING_DEV_x_MIIMON Default: `BONDING_DEV_x_MIIMON='100'`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Gibt an in welchen Zeitabständen (in Millisekunden) die einzelnen Verbindungen eines Bondinggerätes auf Ihren Linkstatus geprüft werden. Es wird also der Linkstatus jedes einzelnen Gerätes dieses Bondinggerätes alle x Millisekunden geprüft. Mit '0' wird die MIIMON Überwachung deaktiviert.

BONDING_DEV_x_USE_CARRIER Default: `BONDING_DEV_x_USE_CARRIER='yes'`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Wenn eine Überwachung des Linkstatus per MIIMON aktiviert wird kann man hier auswählen, ob die Überwachung des Linkstatus durch die netif_carrier_ok() Funktion (bei der Einstellung 'yes') erfolgen soll, oder durch direkte Aufrufe von MII oder ETHTOOL ioctl() Systemaufrufen (mit der Einstellung 'no'). Die netif_carrier_ok() Methode ist effizienter, aber nicht alle Treiber unterstützen diese Methode.

BONDING_DEV_x_UPDELAY Default: `BONDING_DEV_x_UPDELAY='0'`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Der Wert dieser Einstellung multipliziert mit der Einstellung von `BONDING_DEV_x_MIIMON` gibt an nach welcher Zeit eine Verbindung des Bondinggerätes aktiviert wird wenn der entsprechende Link (z.B. ein eth–Gerät) aufgebaut wurde. Damit wird eine Verbindung des Bondinggerätes solange aktiviert, bis der Linkstatus auf 'nicht verbunden' schaltet.

BONDING_DEV_x_DOWNDELAY Default: `BONDING_DEV_x_DOWNDELAY='0'`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Der Wert dieser Einstellung multipliziert mit der Einstellung von `BONDING_DEV_x_MIIMON` gibt an nach welcher Zeit eine Verbindung des Bondinggerätes deaktiviert wird wenn der entsprechende Link (z.B. ein eth–Gerät) ausfällt. Damit wird also eine Verbindung des Bondinggerätes zeitweise deaktiviert, solange bis der Linkstatus wieder auf 'aktiv' schaltet.

BONDING_DEV_x_LACP_RATE Default: `BONDING_DEV_x_LACP_RATE='slow'`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Wenn bei `BONDING_DEV_x_MODE="` der Wert '802.3ad' eingestellt wird, kann man hier angeben wie oft die Linkinformationen mit dem Verbindungspartner (also einem Switch oder einem anderen Linuxrechner) ausgetauscht werden. 'slow' tauscht alle 30 Sekunden die Linkinformationen aus, bei 'fast' werden die Linkinformationen jede Sekunde ausgetauscht.

BONDING_DEV_x_PRIMARY Default: `BONDING_DEV_x_PRIMARY="`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Wenn als Mode 'active-backup' eingestellt bestimmt man hiermit, welches Gerät primär als Ausgabegerät benutzt werden soll. Das ist vor allem sinnvoll, wenn die unterschiedlichen Geräte eine unterschiedliche Geschwindigkeit haben. Ein String (eth0, eth2, etc) der als Primäres Devices verwendet werden soll. Wenn ein Wert eingegeben wird, und das Device ist online, wird es als erstes Ausgabemedium benutzt. Nur wenn das Device offline ist, wird ein anderes Devices benutzt. Andernfalls, sobald ein Ausfall erkannt wird, wird ein neues Standardausgabemedium bestimmt. Dies ist dann praktisch, wenn ein Slave Vorrang gegenüber einem anderen haben soll - wenn bspw. ein Slave 1000 Mbit/s schnell ist und ein anderer 100 Mbit/s. Wenn der 1000 Mbit/s-Slave ausfällt und später wieder hergestellt wurde, kann es von Vorteil sein, dass der schnellere Slave wieder aktiv gesetzt werden kann, ohne beim 100 Mbit/s-Slave künstlich einen Ausfall herbei- zuführen.

BONDING_DEV_x_ARP_INTERVAL Default: `BONDING_DEV_x_ARP_INTERVAL='0'`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Gibt die Frequenz in Millisekunden an nach dem die unter `BONDING_DEV_x_ARP_IP_TARGET_x` angegebenen IP–Adressen (bzw. deren ARP Antwort) geprüft werden. Wenn ARP-Überwachung im Load-Balancing-Mode (mode 0 or 2) genutzt werden soll, sollte der Switch so eingestellt werden, dass er alle Pakete gleich auf alle Verbindungen verteilt - wie etwa Round-Robin. Wenn der Switch so eingestellt ist, dass er die Pakete nach der XOR-Methode verteilt, werden alle Antworten der ARP-Ziele auf der selben Verbindung ankommen und das könnte bei den anderen Team- Mitgliedern zum Ausfall führen. ARP-Überwachung sollte nicht zusammen mit miimon verwandt werden. Wird als Wert 0 übergeben, ist ARP-Überwachung deaktiviert.

BONDING_DEV_x_ARP_IP_TARGETS_N Default: `BONDING_DEV_x_IP_TARGETS_N="`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Die Anzahl der IP–Adressen die für die ARP Prüfung benutzt werden sollen. Es können maximal 16 IP–Adressen überprüft werden.

BONDING_DEV_x_ARP_IP_TARGET_x Default: `BONDING_DEV_x_ARP_IP_TARGET_x="`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Hier wird jeweils eine IP-Adressen angegeben, wenn `BONDING_DEV_x_ARP_INTERVAL` > 0 ist. Diese werden als Ziele der ARP-Anfragen verwandt, die verschickt werden, um die Qualität der Verbindung zu den Zielen festzustellen. Geben sie diese Werte im Format ddd.ddd.ddd.ddd an. Damit ARP-Überwachung funktioniert, muss zumindest eine IP-Adresse angegeben werden.

4.2.2. VLAN - 802.1Q Unterstüzung

Die Unterstützung für VLAN nach 802.1Q ist nur in Verbindung mit entsprechenden Switches sinnvoll. Port-Based VLAN Switche sind *nicht* dafür geeignet. Eine allgemeine Einführung in das Thema VLAN findet sich unter http://www.hoteltravelmovie.de/chrisi/downloads/VLAN_802frame.pdf in deutscher Sprache. Gerade für den Einstieg in das Thema VLAN ist diese Seite geeignet. Auf der Seite http://de.wikipedia.org/wiki/VLAN finden sich auch noch ein wenig Informationen wo man etwas nachlesen kann.

Bitte beachten Sie, dass nicht jede Netzwerkkarte mit VLANs umgehen kann. Einige Netzwerkkarten können überhaupt nicht mit VLANs umgehen, andere benötigen eine angepaßte MTU und einige wenige Karten arbeiten vollkommen problemlos. Der Autor des advanced_networking Paketes benutzt Intelnetzwerkkarten mit dem 'e100' Treiber ohne jedes Problem, eine MTU Anpassung ist nicht notwendig. Der 3COM '3c59x' Treiber benötigt eine MTU Anpassung, die MTU muß auf 1496 eingestellt werden, sonst arbeitet die Karte nicht korrekt. Der 'starfire' Treiber arbeitet nicht korrekt wenn ein VLAN Gerät in einer Bridge aufgenommen wird. In diesem Fall können keine Pakete mehr empfangen werden. Wer also mit VLANs arbeiten will sollte sicherstellen, dass der jeweilige Linux Netzwerkkartentreiber VLANs auch korrekt unterstützt.

OPT_VLAN_DEV Default: `OPT_VLAN_DEV='no'`

Mit `'yes'` wird das VLAN Paket aktiviert, mit `'no'` wird es deaktiviert.

VLAN_DEV_N Default: `VLAN_DEV_N="`

Anzahl der zu konfigurierenden VLAN Geräte.

VLAN_DEV_x_DEV Default: `VLAN_DEV_x_DEV="`

Der Name des Gerätes, das an den VLAN fähigen Switch angeschlossen ist. Das kann z.B. `'eth0'`, `'br1'` oder `'eth2'` sein.

VLAN_DEV_x_VID Default: `VLAN_DEV_x_VID="`

Die VLAN ID, für welches das entsprechende VLAN Gerät erstellt werden soll. Der Name des VLAN Gerätes wird aus dem Prefix 'ethX' und der angehängten VLAN ID (ohne führende '0') erstellt. Wird hier z.B. '42' eingetragen existiert später auf dem fli4l–Router das VLAN Gerät `'eth0.42'`.

Die VLAN Geräte auf dem fli4l–Router heissen immer '<device>.<vid>'. Also wenn ich ein eth–Gerät habe was an einem VLAN–fähigen Switch angeschlossen ist und ich auf dem fli4l–Router die VLANs 10, 11 und 23 nutzen will konfiguriere ich 3 VLAN Geräte mit dem eth–Gerät als `VLAN_DEV_x_DEV='ethX'` und der jeweiligen VLAN ID unter `VLAN_DEV_x_VID="`. Aber wie immer sagt ein Beispiel mehr als tausend Worte, daher hier das passende Beispiel:

```
OPT_VLAN_DEV='yes'
VLAN_DEV_N='3'
VLAN_DEV_1_DEV='eth0'
VLAN_DEV_1_VID='10'# Ergibt device: eth0.10
VLAN_DEV_2_DEV='eth0'
VLAN_DEV_2_VID='11'# Ergibt device: eth0.11
VLAN_DEV_3_DEV='eth0'
VLAN_DEV_3_VID='23'# Ergibt device: eth0.23
```

Bitte immer daran denken die MTU aller beteiligten Geräte zu prüfen. Durch den VLAN Header werden die Frame 4 Bytes länger. Wenn es notwendig ist muss bei den entsprechenden Geräten die MTU auf 1496 geändert werden.

4.2.3. Device MTU - Anpassen der MTU

Unter seltenen Umständen kann es notwendig sein, die MTU eines Gerätes anzupassen. Z.B. einige nicht 100% VLAN kompatible Netzwerkkarten benötigen eine Anpassung der MTU. Bitte denken Sie daran, dass nur wenige Netzwerkkarten in der Lage sind Ethernetframes die größer als 1500 Bytes sind zu verarbeiten!

DEV_MTU_N Default: `DEV_MTU_N="`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Gibt die Anzahl der Geräte an deren MTU geändert werden soll.

DEV_MTU_x Default: `DEV_MTU_x="`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Der Gerätename dessen MTU geändert werden soll gefolgt von der einzustellenden MTU. Beide Angaben werden durch ein Leerzeichen getrennt. Um z.B. für `'eth0'` eine MTU von `'1496'` einzustellen geben Sie folgendes ein:

```
DEV_MTU_N='1'
DEV_MTU_1='eth0 1496'
```

4.2.4. BRIDGE - Ethernet Bridging für fli4l

Hierbei handelt sich um eine vollwertige Ethernet-Bridge, die bei Bedarf nach dem Spanning Tree Protokoll arbeiten kann. Für den Anwender scheint der Rechner an den konfigurierten Ports danach wie ein Layer 3 Switch zu arbeiten.

Weiterführende Informationen zum Thema Bridging finden Sie hier:

Die Homepage des Linux Bridge Projektes: http://bridge.sourceforge.net/.

Die ausführliche und verbindliche Beschreibung des Briding Standards: http://standards.ieee.org/getieee802/download/802.1D-2004.pdf. Vor allem die Informationen ab Seite 153 sind interessant. Bitte beachten Sie, dass der Linux Bridging Code noch nach dem Standard von 1998 arbeitet. Dort gibt es z.B. nur 16 Bit Werte für die Pathcost.

Hier kann man sich die unterschiedlichen Zeitwerte für das Spanning Tree Protocoll berchnen lassen: http://www.dista.de/netstpclc.htm

Wie STP arbeitet kann man auf dieser Seite anhand einiger netter Beispiele sehen: http://web.archive.org/web/20060114052801/http://www.zyxel.com/support/supportnote/ves1012/app/stp.htm

OPT_BRIDGE_DEV Default: `OPT_BRIDGE_DEV='no'`

Mit `'yes'` wird das Bridge Paket aktiviert, mit `'no'` wird es deaktiviert.

BRIDGE_DEV_BOOTDELAY Default: `BRIDGE_DEV_BOOTDELAY='yes'`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Da eine Bridge mindestens 2 × `BRIDGE_DEV_x_FORWARD_DELAY` in Sekunden an Zeit benötigt, um aktiv zu werden, ist diese Zeitspanne abzuwarten, wenn die Devices beim Start von fli4l sofort benötigt werden, um z.B. Syslogmeldungen zu verschicken oder sich per DSL einzuwählen. Wird der Eintrag auf `'yes'` gelassen wird automatisch 2 ×

BRIDGE_DEV_x_FORWARD_DELAY gewartet. Werden die Bridges nicht direkt beim Start benötigt, sollten Sie hier den Wert 'no' eintragen um den Startvorgang des fli4l-Routers zu beschleunigen.

BRIDGE_DEV_N Default: BRIDGE_DEV_N='1'

Die Anzahl der voneinander unabhängigen Bridges. Jede Bridge ist von den anderen vollkommen isoliert zu betrachten. Das gilt insbesondere für die Einstellung von BRIDGE_DEV_x_STP. Es wird pro Bridge ein virtuelles Device mit Namen 'br<nummer>' angelegt.

BRIDGE_DEV_x_NAME Default: BRIDGE_DEV_x_NAME="

Der symbolische Name der Bridge. Dieser Name kann von anderen Paketen benutzt werden um die Bridge unabhängig von dem Gerätenamen zu benutzen.

BRIDGE_DEV_x_DEVNAME Default: BRIDGE_DEV_x_DEVNAME="

Jedes Bridgegerät braucht einen Namen in der Form von 'br<nummer>'. Dabei darf <nummer> eine Zahl zwischen '0' und '99' ohne führende '0' sein. Mögliche Einträge sind also 'br0', 'br9' oder 'br42'. Die Namen können beliebig gewählt werden, die erste Bridge kann also 'br3' heissen und die zweite 'br0'.

BRIDGE_DEV_x_DEV_N Default: BRIDGE_DEV_x_DEV_N='0'

Wie viele Netzwerkgeräte gehören der Bridge an? Die Anzahl der Devices, die fest an die Bridge gebunden werden sollen. Kann auch '0' sein, wenn die Bridge nur als Platzhalter für eine IP-Adresse sein soll, die dann von einem an die Bridge gebundenen VPN-Tunnel übernommen werden soll.

BRIDGE_DEV_x_DEV_x_DEV Gibt an, welches Device an die Bridge gehängt werden kann. hier kann ein eth-Device (z.B. 'eth0'), ein Bonding-Device (z.B. 'bond0') oder auch ein Vlan-Device eingetragen werden (z.B. 'vlan11'). Ein hier eingebundenes Device darf nicht mehr an anderer Stelle verwendet werden und auch keine IP erhalten.

```
BRIDGE_DEV_1_DEV_N='3'
BRIDGE_DEV_1_DEV_1_DEV='eth0.11'#VLAN 11 auf eth0
BRIDGE_DEV_1_DEV_2_DEV='eth2'
BRIDGE_DEV_1_DEV_3_DEV='bond0'
```

BRIDGE_DEV_x_AGING Default: BRIDGE_DEV_x_AGING='300'

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Gibt an nach welcher Zeit alte Einträge in der MAC Tabelle der Bridge gelöscht werden. Wenn die hier angegebene Zeit in Sekunden keine Daten von dem Rechner mit der Netzwerkkarte empfangen oder verschickt worden sind wird diese entsprechende MAC Adresse aus der Bridge MAC Tabelle entfernt.

BRIDGE_DEV_x_GARBAGE_COLLECTION_INTERVAL Default: BRIDGE_DEV_x_GARBAGE_COLLECTION_IN

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Gibt an nach wieviel Sekunden eine „Müllsammlung“ gemacht wird. Dabei werden alle dynamischen Einträge der Bridge auf ihre Gültigkeit geprüft und veraltete oder nicht mehr gültige Einträge entfernt. Insbesondere bedeutet dies, dass alte nicht mehr gültige Verbindungen entfernt werden.

BRIDGE_DEV_x_STP Default: `BRIDGE_DEV_x_STP='no'`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Das Spanning Tree Protokoll erlaubt es, mehrere Verbindungen zu anderen Switches zu unterhalten. Dadurch wird eine Redundanz erziehlt, die die Funktionsfähigkeit des Netzwerkes im Falle eines Ausfalls einer Leitung gewährleistet. Ohne den Einsatz von STP sind redundante Leitungen zwischen Switches nicht möglich, dass Netzwerk würde nicht funktionieren. Das STP versucht immer die schnellstmögliche Verbindung zwischen zwei Switches zu benutzen, so dass auch der Einsatz von zwei unterschiedlich schnellen Leitungen sinnvoll ist. So könnte man z.B. eine 1 Gbit/s Verbindung als Hauptverbindung benutzen und eine zweite 100 Mbit/s Verbindung als Sicherheit.

Einen guten Artikel mit einigen Hintergrundinformationen finden Sie auf dieser Seite: http://de.wikipedia.org/wiki/Spanning_Tree_Protocol.

BRIDGE_DEV_x_PRIORITY Default: `BRIDGE_DEV_x_PRIORITY="`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Nur gültig wenn `BRIDGE_DEV_x_STP='yes'` gesetzt wird!

Welche Priorität hat diese Bridge? Die Bridge mit der geringsten Priorität in der aktuellen Landschaft gewinnt die Wahl zur Hauptbridge. Jede Bridge sollte eine unterschiedliche Priorität haben. Bitte beachten Sie, dass die Bridge mit der geringsten Priorität auch über die größte Bandbreite verfügen sollte, da diese alle 2 Sekunden (oder die unter `BRIDGE_DEV_x_HELLO`) Steuerpakete verschickt und auch der gesamte restliche Datenverkehr über sie abgewickelt wird.

Gültige Werte sind `'0'` bis `'61440'` in Schritten von 4096.

BRIDGE_DEV_x_FORWARD_DELAY Default: `BRIDGE_DEV_x_FORWARD_DELAY='15'`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Nur gültig wenn `BRIDGE_DEV_x_STP='yes'` gesetzt wird!

Wenn ein Bridgeanschluß deaktiviert war und erneut aktiviert werden soll, oder auch wenn der Anschluß gerade neu zur Bridge hinzugekommen ist, dauert es die angegebene Zeitspanne (in Sekunden) × 2 bis der Anschluß Daten weiterleiten kann. Dieser Parameter ist maßgebend für die Dauer, die die Bridge benötigt um einer tote Verbindung zu erkennen. Die Zeitspanne wird nach folgender Formel in Sekunden berechnet:

`BRIDGE_DEV_x_MAX_MESSAGE_AGE`+(2 × `BRIDGE_DEV_x_FORWARD_DELAY`)

Daraus ergibt sich mit den Standardwerten: $20 + (2 \times 15) = 50$ Sekunden. Die Zeit bis eine tote Verbindung erkennt wird kann minimiert werden, wenn `BRIDGE_DEV_x_HELLO` auf 1 Sekunde und die `BRIDGE_DEV_x_FORWARD_DELAY` auf 4 Sekunden gestellt wird. Zusätzlich muss `BRIDGE_DEV_x_MAX_MESSAGE_AGE` auf 4 Sekunden eingestellt werden. Daraus ergibt sich folgender Wert: $4 + (2 \times 4) = 12$ Sekunden. Schneller geht es nicht.

BRIDGE_DEV_x_HELLO Default: `BRIDGE_DEV_x_HELLO='2'`

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Nur gültig wenn `BRIDGE_DEV_x_STP='yes'` gesetzt wird!

Die mit BRIDGE_DEV_x_HELLO angegeben Zeit ist der Zeitabstand in Sekunden, in dem die sogenannten Hello Nachrichten von der Hauptbridge verschickt werden. Diese Nachrichten sind für die automatische Konfiguration von STP notwendig.

BRIDGE_DEV_x_MAX_MESSAGE_AGE Default: BRIDGE_DEV_x_MAX_MESSAGE_AGE='20'

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Nur gültig wenn BRIDGE_DEV_x_STP='yes' gesetzt wird!

Die maximale Gültigkeitsdauer der letzten Hello Nachricht. Wenn innerhalb dieser Zeit (in Sekunden) keine neue Hello Nachricht empfangen wird, wird eine neue Wahl der Hauptbridge ausgelöst. Deshalb darf dieser Wert **nie** kleiner als 2 × BRIDGE_DEV_x_HELLO sein.

BRIDGE_DEV_x_DEV_x_PORT_PRIORITY Default: BRIDGE_DEV_x_DEV_x_PORT_PRIORITY='128'

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Nur gültig wenn BRIDGE_DEV_x_STP='yes' gesetzt wird!

Ist nur relevant, wenn mehrere Verbindungen mit der selben BRIDGE_DEV_x_DEV_x_PATHCOST zum selben Ziel führen. Ist dies der Fall wird die Verbindung mit der geringsten Priorität ausgewählt.

Gültige Werte sind '0' bis '240' in Schritten von '16'.

BRIDGE_DEV_x_DEV_x_PATHCOST Default: BRIDGE_DEV_x_DEV_x_PATHCOST='100'

Diese Einstellung ist optional und kann auch komplett weggelassen werden.

Nur gültig wenn BRIDGE_DEV_x_STP='yes' gesetzt wird!

Bestimmt indirekt die Bandbreite für diese Verbindung. Je geringer der Wert, desto höher ist die Bandbreite und damit wird die Verbindung höher priorisiert.

Die vorgeschlagene Berechnungsgrundlage ist 1000000 / kbit/s, was zu den in Tabelle 4.1 aufgelisteten Werten führt. Bitte beachten Sie, dass bei der Berechnung die tatsächlich nutzbare Bandbreite in die Formel eingesetzt werden muss. Dadurch ergeben sich vor allem für WLAN deutliche niedrigere Werte als man erwarten würde.

Hinweis: Der aktuelle IEEE Standard von 2004 benutzt für die Bandbreitenberechnung 32 Bitzahlen, die von Linux noch nicht unterstützt werden.

Bandbreite	Einstellung von BRIDGE_DEV_x_DEV_x_PATHCOST
64 kbit/s	15625
128 kbit/s	7812
256 kbit/s	3906
10 Mbit/s	100
11 Mbit/s	190
54 Mbit/s	33
100 Mbit/s	10
1 Gbit/s	1

Tabelle 4.1.: Werte für BRIDGE_DEV_x_DEV_x_PATHCOST in Abhängigkeit von der Bandbreite

4.2.5. Anmerkungen

Eine Bridge leitet jede Art von Ethernetdaten weiter - somit läßt sich z.B. auch ein normales DSL-Modem über WLAN ansprechen als hätte es eine WLAN Schnittstelle. Es wird kein Paket, welches die Bridge passiert auf irgendwelche unerwünschten Aktivitäten hin untersucht (das heisst der fli4l Paketfilter wird nicht aktiv!), wodurch der Einsatz z.B. als WLAN Access Point nur unter sorgfältiger Abwägung der Sicherheitsrisiken zu empfehlen ist. Es gibt allerdings auch die Möglichkeit die EBTables Unterstützung zu aktivieren.

4.2.6. EBTables - EBTables für Fli4l

Ab der Version 2.1.9 hat fli4l auch rudimentäre EBTables Unterstützung. Mit `OPT_EBTABLES='yes'` wird die EBTables Unterstützung aktiviert. Damit werden alle EBTables Kernelmodule geladen und das ebtables Programm auf dem fli4l–Router zur Verfügung gestellt. Im Gegensatz zur wesentlich vereinfachten Konfiguration des netfilter durch die verschiedenen Filterlisten von fli4l ist es notwendig selbstständig ein ebtables Skript zu schreiben. Das bedeutet, sie müssen das komplette ebtables Skript selbst schreiben.

Für Hintergrundinformationen zu der EBTables Unterstützung lesen sie bitte die Dokumentation auf der EBTables Homepage unter http://ebtables.sourceforge.net.

Es gibt die Möglichkeit ebtables Kommandos vor und nach dem Einrichten des netfilters (die `PF_INPUT_x`, `PF_FORWARD_x` usw) auf dem fli4l–Router abzusetzen. Legen Sie dazu je nach Bedarf im Verzeichnis config/ebtables die Dateien ebtables.pre und ebtables.post an. Die Datei ebtables.pre wird vor der Konfiguration des netfilters ausgeführt, die ebtables.post Datei danach. Bitte bedenken sie, dass ein Fehler in den ebtables Skripten unter Umständen den Startvorgang des fli4l–Routers unterbricht!

Vor Einsatz der EBTables Unterstützung sollten sie unbedingt die komplette EBTables Dokumentation lesen. Durch den Einsatz von EBTables ändert sich das Verhalten des fli4l–Router unter Umständen! So funktioniert z.B. das mac: Filtern in der PF_FORWARD nicht mehr wie gewohnt.

Sehr interessant ist folgende Seite, die einen kleinen Einblick in die Funktionsweise der EBTables Unterstützung bringt: http://ebtables.sourceforge.net/br_fw_ia/br_fw_ia.html.

4.2.7. Beispiel

Für das Verständnis ist ein einfaches Beispiel sicher hilfreich. Wir gehen in unserem Beispiel von 2 Gebäudeteile aus, die mit 2 x 100 Mbit/s Verbindungen verbunden sind. Es sollen darüber 4 getrennte Netze von einem Gebäude in das andere geroutet werden.

Um dies zu verwirklichen, bietet sich eine Kombinatinon aus Bonding (die 2 vorhandenen 100 Mbit/s Verbindungen in deren Übertragungsleistung zusammenfassen), VLAN (um mehrere getrennte Netze auf einer zusammengefassten Leitung transportieren zu können) und Bridging (um die Netze in den Gebäuden in das Bond/VLAN Gebilde einhängen zu können. (erfolgreich getestet mit 2x Intel e100 Karten und 1x Adaptec 4-Port Karte ANA6944.) Die beiden e100 haben in diesem Beispiel die Gerätenamen `'eth0'` und `'eth1'` und werden für die Gebäudeverbindung verwendet. Aktuell sind uns keine anderen Kartentypen außer die Intel e100 bekannt, die reibungslos mit VLAN zusammenarbeiten. Gigabit-Karten sollten aber prinzipiell auch funktionieren. Die 4 Anschlüsse der Multiportkarte dienen für die jeweiligen Netzwerke und haben die Gerätenamen `'eth2'` bis `'eth5'`.

Zuerst werden die beiden 100 Mbit/s Strecken gebondet:

```
OPT_BONDING_DEV='yes'
BONDING_DEV_N='1'
BONDING_DEV_1_DEVNAME='bond0'
BONDING_DEV_1_MODE='balance-rr'
BONDING_DEV_1_DEV_N='2'
BONDING_DEV_1_DEV_1='eth0'
BONDING_DEV_1_DEV_2='eth1'
```

Dadurch wir das Gerät `'bond0'` erzeugt. Auf diese Bondingverbindung werden jetzt die VLANs aufgebaut, wir verwenden im Beispiel die VLAN-IDs 11, 22, 33 und 44:

```
OPT_VLAN_DEV='yes'
VLAN_DEV_N='4'
VLAN_DEV_1_DEV='bond0'
VLAN_DEV_1_VID='11'
VLAN_DEV_2_DEV='bond0'
VLAN_DEV_2_VID='22'
VLAN_DEV_3_DEV='bond0'
VLAN_DEV_3_VID='33'
VLAN_DEV_4_DEV='bond0'
VLAN_DEV_4_VID='44'
```

Und über diese VLAN Verbindungen wird nun eine Bridge in die einzelnen Netzwerksegmente gelegt. Ein Routing ist somit nicht notwendig.

```
OPT_BRIDGE_DEV='yes'
BRIDGE_DEV_N='4'
BRIDGE_DEV_1_NAME='_VLAN11_'
BRIDGE_DEV_1_DEVNAME='br11'
BRIDGE_DEV_1_DEV_N='2'
BRIDGE_DEV_1_DEV_1='bond0.11'
BRIDGE_DEV_1_DEV_2='eth2'
BRIDGE_DEV_2_NAME='_VLAN22_'
BRIDGE_DEV_2_DEVNAME='br22'
BRIDGE_DEV_2_DEV_N='2'
BRIDGE_DEV_2_DEV_1='bond0.22'
BRIDGE_DEV_2_DEV_2='eth3'
BRIDGE_DEV_3_NAME='_VLAN33_'
BRIDGE_DEV_3_DEVNAME='br33'
BRIDGE_DEV_3_DEV_N='2'
BRIDGE_DEV_3_DEV_1='bond0.33'
BRIDGE_DEV_3_DEV_2='eth4'
BRIDGE_DEV_4_NAME='_VLAN44_'
BRIDGE_DEV_4_DEVNAME='br44'
BRIDGE_DEV_4_DEV_N='2'
BRIDGE_DEV_4_DEV_1='bond0.44'
BRIDGE_DEV_4_DEV_2='eth5'
```

Damit sind jetzt alle 4 Netze vollkommen transparent miteinander verbunden und teilen sich die 200 Mbit/s Verbindung. Selbst bei Ausfall einer 100 Mbit/s Verbindung funktioniert die

Verbindung noch. Bei Bedarf kann auch noch die EBTablesunterstützung aktiviert werden um z.B. bestimmte Paketfilter zu aktivieren.

Diese Konfiguration wird auf zwei fli4l-Routern eingerichtet. Ich denke dieses Beispiel zeigt eindrucksvoll welche Möglichkeiten das advanced_networking Paket ermöglicht.

4.3. CHRONY - Network Time Protocol Server/Client

OPT_CHRONY erweitert fli4l um das Network Time Protocol (Seite 100) (NTP). Dies ist nicht mit dem *normalen* Time Protokoll zu verwechseln, welches das alte OPT_TIME bereitstellt. Die Protokolle sind nicht kompatibel und somit werden gegebenenfalls neue Client-Programme, die NTP beherrschen, benötigt. Falls man nicht auf das einfache Time Protokoll verzichten kann, so läßt sich dieses Protokoll zusätzlich aktivieren.
OPT_CHRONY arbeitet sowohl im Server, als auch im Client Modus. In der Funktion des Client gleicht es die Zeit des fli4l mit Zeitreferenzen (Time Server) im Internet ab. In der Grundeinstellung nutzt OPT_CHRONY bis zu drei verschiedene Time Server aus dem Fundus von pool.ntp.org (Seite 100). Es ist jedoch auch möglich, über die Konfigurationsdatei eine andere Auswahl an Time Servern zu treffen. So ist es beispielsweise sinvoll Server aus Europa zu wählen. Möglich ist das, indem man als Server de.pool.ntp.org angibt, wenn der Router bzw. der Provider in Deutschland ist. Weitere Informationen dazu auf der Webseite von pool.ntp.org (Seite 100).

In der Funktion des Server dient OPT_CHRONY als Zeitreferenz für das lokale Netzwerk (LAN). NTP arbeitet auf Port 123.

Chrony zeichnet sich dadurch aus, daß es keine fortlaufende Verbindung zum Internet benötigt. Sobald die Verbindung getrennt wird (offline), erhält chrony hiervon Kenntnis und stellt den Zeitabgleich mit den Internet Time Servern ein. Somit löst chrony keinen neuen Verbindungsaufbau aus. Weiterhin verhindert chrony nicht die automatische Verbindungsunterbrechung, falls die `HUP_TIMEOUT`, also die Zeit, in der keine Daten mit dem Internet ausgetauscht werden, erreicht wurde.

Damit der Zeitabgleich reibungslos funktioniert, sollte folgendes beachtet werden:

- Chrony erwartet, daß die BIOS-Uhr in der Zeitzone UTC läuft. Falls nicht, muß dies in der Konfigurationsdatei geändert werden!
  UTC = Deutsche Zeit minus 1 (Winter) bzw. 2 (Sommer) Stunde(n)
- Seit der Version 2.1.12 setzt Chrony die Uhrzeit mit der ersten Verbindung zum Internet korrekt, auch wenn der Zeitunterschied sehr groß sein sollte (beispielsweise bei defekter Mainboardbatterie).
- Sollte das BIOS Jahreszahlen nach 1999 nicht richtig darstellen können (Year 2000 Bug) bzw. die Implementation der BIOS Uhr fehlerhaft sein, so muß `OPT_Y2K='yes'` (Seite 83) aktiviert werden!

Es können nur Time Server im Internet über die Default Route (`0.0.0.0/0`) erreicht werden, da nur die Default Route Chrony in den online Zustand versetzt. Ist der Router als LAN-Router konfiguriert, also keine DSL oder ISDN Circuits definiert, ist Chrony permanent im online Zustand.

Disclaimer: *Der Autor gibt weder eine Garantie auf die Funktionsfähigkeit des OPT_-CHRONY, noch haftet er für Schäden, z.B. Datenverlust, die durch den Einsatz von OPT_-CHRONY entstehen.*

4.3.1. Konfiguration des OPT_CHRONY

Die Konfiguration erfolgt, wie bei allen fli4l Opts, durch Anpassung der Datei `Pfad/fli4l-3.6.2/<config>/chrony.txt` an die eigenen Anforderungen. Jedoch sind fast alle Variablen des OPT_CHRONY optional. Optional heißt, die Variablen können, müssen aber nicht in der Konfigurationsdatei auftauchen. Somit ist die chrony Konfigurationsdatei im Auslieferungszustand fast leer und die optionalen Variablen sind sinnvoll vorbelegt. Möchte man dennoch eine anderen Konfiguration, müssen die Variablen von Hand eingefügt werden. Im weiteren erfolgt nun die Beschreibung der einzelnen Variablen:

OPT_CHRONY Default: `OPT_CHRONY='no'`

Die Einstellung `'no'` deaktiviert das OPT_CHRONY vollständig. Es werden keine Änderungen an der fli4l Bootdiskette bzw. dem Archiv `opt.img` vorgenommen. Weiter überschreibt das OPT_CHRONY grundsätzlich keine anderen Teile der fli4l Installation, mit einer Ausnahme. Es wird die Filterdatei ausgetauscht, die dafür sorgt, das Anfragen von außen nicht als Traffic gewertet werden (fli4l legt sicher nach erreichen der Hangup Time auf). Die neue Filterdatei legt fest, daß der chrony-Traffic ebenfalls nicht mitgezählt wird, somit legt der Router weiterhin sicher auf.
Um OPT_CHRONY zu aktivieren, ist die Variable `OPT_CHRONY` auf `'yes'` zu setzen.

CHRONY_TIMESERVICE Default: `CHRONY_TIMESERVICE='no'`

Mit `CHRONY_TIMESERVICE` kann ein weiteres Protokoll zur Zeitübermittlung aktiviert werden. Dieses ist nur dann nötig, wenn die lokalen Rechner nicht mit NTP arbeiten können. Das zusätzliche Protokoll ist RFC 868 konform und arbeitet auf Port 37. Wenn immer möglich, sollte NTP vorgezogen werden.

Einen herzlichen Dank an Christoph Schulz, der das Programm `srv868` beigesteuert hat.

CHRONY_TIMESERVER_N Default: `CHRONY_TIMESERVER_N='3'`

`CHRONY_TIMESERVER_N` legt die Anzahl der als Referenz benutzten Time Server fest. Der Anzahl entsprechend sind `CHRONY_TIMESERVER_x` Variablen anzulegen. Der Index `x` muß fortlaufend bis zur Gesamtanzahl heraufgezählt werden.
In der Grundeinstellung nutzt chrony drei Internet Time Server aus dem Fundus von pool.ntp.org (Seite 100).

CHRONY_TIMESERVER_x Default: `CHRONY_TIMESERVER_x='pool.ntp.org'`

Mit `CHRONY_TIMESERVER_x` kann eine eigene Liste von Internet Time Servern angelegt werden. Die Time Server können sowohl durch ihre IP als auch über ihren DNS Namen spezifiziert werden.

CHRONY_LOG Default: `CHRONY_LOG='/var/run'`

`CHRONY_LOG` bezeichnet das Verzeichnis, indem chrony Information über die BIOS Uhr und die Zeitkorrektur ablegt. Sollte in der Regel nicht verändert werden.

CHRONY_BIOS_TIME Default: `CHRONY_BIOS_TIME='utc'`

Damit chrony die Zeit der BIOS Uhr (RTC = real time clock) richtig auswerten kann, wird mittels `CHRONY_BIOS_TIME` übermittelt, ob die Uhr auf lokaler `'local'` oder universaler Zeit `'utc'` (UTC - Universal Coordinated Time) läuft.

4.3.2. Support

Support wird nur im Rahmen der fli4l Newsgroups (Seite 100) geleistet.

4.3.3. Literatur

Homepage von chrony: http://chrony.tuxfamily.org/
NTP: The Network Time Protocol: http://www.ntp.org/
pool.ntp.org: public ntp time server for everyone: http://www.pool.ntp.org/de/
RFC 1305 - Network Time Protocol (Version 3) Specification, Implementation: http://www.faqs.org/rfcs/rfc1305.html
fli4l Newsgroups und ihre Spielregeln: http://www.fli4l.de/hilfe/newsgruppen/

4.4. DHCP_CLIENT - Dynamic Host Configuration Protocol

Mit Hilfe dieses Paketes kann der Router IP-Adressen für seine Interfaces dynamisch beziehen. Das Paket und seine Parameter werden im Folgenden erklärt.

4.4.1. OPT_DHCP_CLIENT

Ein DHCP-Client kann verwendet werden, um eine IP-Adresse für ein oder mehrere Interface(s) des Routers zu beziehen - dies erfolgt meist vom Service Provider. Diese Möglichkeit der Anbindung kommt derzeit hauptsächlich bei Kabelmodem-Betreibern und in der Schweiz, in den Niederlanden und in Frankreich vor. Manchmal wird diese Konfiguration auch benötigt, wenn der Router hinter einem anderen Router eingebunden wird, der die Adressen per DHCP verteilt (z.B. FritzBox).

Beim Start des Routers wird für die angegebenen Interfaces eine IP-Adresse bezogen. Anschließend wird diese dem Interface zugewiesen und bei Bedarf die Default- Route auf dieses Interface gelegt.

OPT_DHCP_CLIENT Muss auf 'yes' gesetzt werden, wenn einer der DHCP-Clienten verwendet werden soll.

Standard-Einstellung: `OPT_DHCP_CLIENT='no'`

DHCP_CLIENT_TYPE Das Paket kommt momentan mit zwei verschiedenen Implementierungen eines DHCP-CLienten, dem dhclient und dem dhcpcd. Hier kann man auswählen, welchen man verwenden möchte.

Standard-Einstellung: `DHCP_CLIENT_TYPE='dhcpcd'`

DHCP_CLIENT_N Hier wird die Anzahl zu konfigurierender Interfaces angegeben.

DHCP_CLIENT_x_IF Hier wird das zu konfigurierende Interface als Referenz auf `IP_NET_x_DEV` angegeben, z.B. `DHCP_CLIENT_1_IF='IP_NET_1_DEV'`. Der dhcp-client entnimmt der entsprechenden Variable das passende Device. In der base.txt sollte als Platzhalter anstelle einer IP-Adresse mit Netzmaske '*dhcp*' eingetragen werden.

DHCP_CLIENT_x_ROUTE Hier kann angegeben werden, ob und wie über das Interface eine Route konfiguriert werden soll. Die Variable kann auf folgende Werte gesetzt werden:

none Es wird keine Route über das Interface gelegt.

default Es wird eine default-Route über das Interface gelegt.

imond Der Imond managed die default-Route für dieses Interface.

Standard-Einstellung: `DHCP_CLIENT_x_ROUTE='default'`

DHCP_CLIENT_x_USEPEERDNS Wird dieser Parameter auf 'yes' gesetzt und über das Device eine default-Route gelegt, so wird der vom ISP über- gebene DNS als Forwarder für den DNS auf dem Router verwendet - dieser muss dafür natürlich aktiviert werden - siehe base.txt.

Standard-Einstellung: `DHCP_CLIENT_x_USEPEERDNS='no'`

DHCP_CLIENT_x_HOSTNAME Manche Provider verlangen eine Übermittlung eines Hostnamens. Dieser Name ist vom Provider zu erfahren und hier anzugeben. Es muß nicht identisch mit dem Hostnamen des Routers sein.

Standard-Einstellung: `DHCP_CLIENT_x_HOSTNAME=''`

DHCP_CLIENT_DEBUG Liefert weitere Informationen, wenn eine Adresse angefordert wird.

Standard-Einstellung: weglassen oder `DHCP_CLIENT_DEBUG='no'`

4.5. DNS_DHCP - Hostnamen, DNS- und DHCP-Server sowie DHCP-Relay

Das Paket dns_dhcp stellt die folgenden Funktionen zur Verfügung:

1. Verwaltung von Hosts
2. DNS-Server
3. DHCP-Server
4. DHCP-Relay
5. TFTP-Server

4.5.1. Hostnamen

Hosts

HOSTS_N HOST_x_{attribute} Es sollten alle Rechner im LAN beschrieben werden - mit IP-Adresse, Namen, Aliasnamen und evtl. Mac-Adressen für die dhcp-Konfiguration . Dazu setzt man zunächst die Anzahl der Rechner mit der Variablen `HOSTS_N`.

Hinweis: Seit Version 3.4.0 wird der Eintrag für den Router aus den Angaben in der `<config>/base.txt` generiert. Sollen zusätzliche Aliasnamen aufgenommen werden, siehe auch `HOSTNAME_ALIAS_N` (Seite 77).

Anschließend werden mit den Attributen die Eigenschaften des Hostes definiert. Dabei sind einige Attribute Pflicht, wie z.B. IP-Adresse und Name, die anderen optional, d.h. man kann, aber man muß sie nicht spezifizieren.

NAME – Name des n-ten Hostes

IP4 – IP-Adresse (ipv4)des n-ten Hostes

IP6 – IP-Adresse (ipv6)des n-ten Hostes (optional)

DOMAIN – DNS-Domain des n-ten Hostes (optional)

ALIAS_N – Anzahl der Alias-Namen des n-ten Hostes

ALIAS_m – m-ter Alias-Name für den n-ten Host

MAC – Mac Adresse des n-ten Hostes

DHCPTYP – Vergabe der IP-Adresse per DHCP abhängig von MAC oder NAME (optional)

In der Beispiel-Datei sind 4 Rechner konfiguriert - nämlich die PCs “client1”, “client2”, “client3” und “client4”.

```
HOST_1_NAME='client1'                # 1st host: ip and name
HOST_1_IP4='192.168.6.1'
```

Aliasnamen müssen mit kompletter Domain angegeben werden.

Die MAC-Adresse ist optional und ist nur dann relevant, wenn fli4l zusätzlich als DHCP-Server eingesetzt wird. Dies wird in der Beschreibung zum optionalen Programmpaket “OPT_DHCP” erklärt, siehe unten. Ohne Einsatz als DHCP-Server sind lediglich die IP-Adresse, der Name des Rechners und eventuell Aliasnamen einzusetzen. Die MAC-Adresse ist eine 48-Bit-Adresse und besteht aus 6 Hex-Werten, welche durch einen Doppelpunkt voneinander getrennt werden, z.B.

```
HOST_2_MAC='de:ad:af:fe:07:19'
```

Hinweis: Wird fli4l um das IPv6-Paket ergänzt, brauchen keine IPv6-Adressen hinterlegt zu werden, wenn gleichzeitig die MAC-Adressen der Hosts vorliegen, weil das IPv6-Paket dann die IPv6-Adressen automatisch berechnet (modifiziertes EUI-64). Natürlich kann man aber den Automatismus unterbinden und feste IPv6-Adressen vorgeben, wenn man dies wünscht.

Extra Hosts

HOSTS_EXTRA_N HOST_EXTRA_x_NAME HOST_EXTRA_x_IP4 HOST_EXTRA_x_IP6
Mit diesen Variablen können weitere Hosts hinzugefügt werden die nicht der lokalen Domain angehören wie z.b. Hosts die sich auf der anderen Seite eines VPNs befinden.

4.5.2. DNS-Server

OPT_DNS Um den DNS-Server zu aktivieren ist die Variable `OPT_DNS` mit ‘yes‘ zu belegen.

Werden im LAN keine Windows-Rechner verwendet oder ist bereits ein DNS-Server vorhanden, kann man `OPT_DNS` auf ‘no’ setzen und den Rest in diesem Abschnitt übergehen.

Im Zweifel immer (Standard-Einstellung): `OPT_DNS`=’yes’

Allgemeine DNS-Optionen

DNS_LISTENIP_N DNS_LISTENIP_x Wenn Sie `OPT_DNS`=’yes’ gewählt haben, können Sie mit Hilfe von `DNS_LISTENIP_N` die Anzahl, und mit `DNS_LISTENIP_1` bis `DNS_LISTENIP_N` lokale IPs angeben, auf denen `dnsmasq` DNS-Anfragen annehmen darf. Sollten Sie bei `DNS_LISTENIP_N` eine 0 eingetragen haben, beantwortet `dnsmasq` DNS-Anfragen auf allen lokalen IPs. An dieser Stelle dürfen nur IPs von existierenden Schnittstellen (ethernet, wlan ...) verwendet werden, es kommt sonst zu Warnmeldungen beim Start des Routers. Alternativ ist nun möglich hier auch ALIAS-Namen zu verwende, z. B. `IP_NET_1_IPADDR`

Im Zweifelsfalle können die Standardeinstellungen übernommen werden.

DNS_VERBOSE Logging von DNS-Queries: ‘yes’ oder ‘no’

Für ausführlichere Ausgaben des DNS, muß `DNS_VERBOSE` auf yes gesetzt werden. In diesem Fall werden DNS-Anfragen an den Nameserver protokolliert - und zwar über die syslog-Schnittstelle. Damit die Ausgaben auch sichtbar werden, ist dann auch die Variable `OPT_SYSLOGD=’yes’` (Seite 81) zu setzen, s.u.

DNS_MX_SERVER Mit dieser Variable gibt man hier den Hostnamen für den MX-Record (Mail-Exchanger) für die in `DOMAIN_NAME` definierte Domain an. Ein MTA (Mail-Transport-Agent, wie z.B. sendmail) auf einem internen Server fragt per DNS nach einem Mail-Exchanger für die Zieldomain der zuzustellenden Mail. Der DNS-Server liefert hiermit dem MTA den entsprechenden Host, der für Mails der Domain `DOMAIN_NAME` zuständig ist.

Dies ist keine automatische Konfiguration für Mail-Clients, wie z.B. Outlook! Also bitte nicht gmx.de hier eintragen und dann wundern, warum Outlook nicht funktioniert.

DNS_FORBIDDEN_N DNS_FORBIDDEN_x Hier können Sie Domains angeben, bei denen DNS-Queries vom DNS-Server prinzipiell als “nicht vorhanden” beantwortet werden sollen.

Beispiel:

```
DNS_FORBIDDEN_N=’1’
DNS_FORBIDDEN_1=’foo.bar’
```

In diesem Fall wird zum Beispiel eine Anfrage nach www.foo.bar mit einem Fehler beantwortet.

Man kann damit auch ganze Top-Level-Domains verbieten:

```
DNS_FORBIDDEN_1='de'
```

Dann ist die Namensauflösung für sämtliche Rechner in der DE-Topleveldomain abgeschaltet.

DNS_REDIRECT_N DNS_REDIRECT_x DNS_REDIRECT_x_IP Hier können Domains angegeben werden, bei welchen DNS-Queries vom DNS-Server auf eine spezielle IP umgeleitet werden.

Beispiel:

```
DNS_REDIRECT_N='1'
DNS_REDIRECT_1='yourdom.dyndns.org'
DNS_REDIRECT_1_IP='192.168.6.200'
```

In diesem Fall wird zum Beispiel eine Anfrage nach yourdom.dyndns.org mit der IP 192.168.6.200 beantwortet. Somit kann man externe Domains auf andere IPs umleiten.

DNS_BOGUS_PRIV Setzt man diese Variable auf 'yes', werden reverse-lookups für IP-Adressen nach RFC1918 (Private Address Bereiche) nicht vom dnsmasq an andere DNS-Server weitergeleitet, sondern vom dnsmasq beantwortet.

DNS_FILTERWIN2K setzt man diese Variable auf 'yes' werden alle Windows-Spezifischen Namensanfragen gefiltert sowie alle SRV requests gefiltert/geblockt.

DNS_SUPPORT_IPV6 (optional)

setzt man diese optionale Variable auf 'yes' wird die Unterstützung für IPV6- Adressen des DNS-Servers aktiviert.

Spezielle DNS-Server

DNS_SPECIAL_N DNS_SPECIAL_x Es gibt besondere Situationen, wo die Angabe eines oder mehrerer DNS-Server sinnvoll ist, z.B. bei Einsatz von fli4l im Intranet ohne Internet-Anschluss oder einem Mix von diesen (Intranet mit eigenem DNS–Server und zusätzlich Internet-Anschluss).

Stellen wir uns folgendes Szenario vor:

- Circuit 1: Einwahl in das Internet
- Circuit 2: Einwahl in das Firmen-Netz 192.168.1.0 (firma.de)

Dann wird man `ISDN_CIRC_1_ROUTE` auf '0.0.0.0' und `ISDN_CIRC_2_ROUTE` auf '192.168.1.0' setzen. Bei Zugriff auf Rechner mit IP-Adresse 192.168.1.x wird fli4l dann den Circuit 2, sonst den Circuit 1 benutzen. Wenn das Firmen-Netz aber nicht öffentlich ist, wird in diesem vermutlich ein eigener DNS-Server betrieben. Nehmen wir an, die Adresse dieses DNS-Servers wäre 192.168.1.12 und der Domain-Name wäre "firma.de".

In diesem Fall gibt man an:

```
DNS_SPECIAL_N='1'
DNS_SPECIAL_1_DNSIP='192.168.1.12'
DNS_SPECIAL_1_DOMAIN='firma.de''
```

Dann werden bei DNS-Queries von xx.firma.de der firmeninterne DNS-Server, sonst die DNS-Server im Internet verwendet.

Ein anderer Fall:

- Circuit 1: Internet
- Circuit 2: Firmen-Netz 192.168.1.0 *mit* Internet-Anschluss

Hier hat man also die Möglichkeit, auf 2 Wegen in das Internet zu gelangen. Möchte man Geschäftliches und Privates trennen, bietet sich dann folgendes an:

```
ISDN_CIRC_1_ROUTE='0.0.0.0'
ISDN_CIRC_2_ROUTE='0.0.0.0'
```

Man legt also auf beide Circuits eine Default-Route und schaltet dann die Route mit dem imond-Client um - je nach Wunsch. Auch in diesem Fall sollte man `DNS_SPECIAL_N` und `DNS_SPECIAL_x` wie oben beschrieben einstellen.

Möchte man auch die Reverse-DNS-Auflösung für ein so erreichbares Netz nutzen, z.B. wird ein Reverse-Lookup von einigen Mail-Server gemacht, gibt man in der optionalen Variable `DNS_SPECIAL_x_NETWORK`, das/die Netz(werke) an, für die der Reverse-Lookup aktiviert werden soll. Das folgende Beispiel verdeutlicht das:

```
DNS_SPECIAL_N='2'
DNS_SPECIAL_1_DNSIP='192.168.1.12'
DNS_SPECIAL_1_NETWORK='192.168.1.0/24'
DNS_SPECIAL_2_DNSIP='192.168.2.12'
DNS_SPECIAL_2_DOMAIN='bspfirma.de'
DNS_SPECIAL_2_NETWORK='192.168.2.0/24 192.168.3.0/24'
```

DNS_REBINDOK_N DNS_REBINDOK_x_DOMAIN Der Nameserver *dnsmasq* lehnt normalerweise Antworten anderer Nameserver ab, die IP-Adressen aus privaten Netzwerken enthalten. Er verhindert dadurch eine bestimmte Klasse von Angriffen auf das Netzwerk. Hat man allerdings eine Domain in einem Netzwerk mit privaten IP-Adressen und einen extra Nameserver, der für dieses Netz zuständig ist, liefert der genau die Antworten, die vom *dnsmasq* abgelehnt werden würden. Diese Domains kann man in `DNS_REBINDOK_x` auflisten, die entsprechenden Antworten auf Anfragen zu der Domain werden dann akzeptiert. Ein weiteres Beispiel für Nameserver, die private IP-Adressen als Antwort liefern, sind sogenannte "Real-Time Blacklist Server". Ein Beispiel basierend auf diesen könnte wie folgt aussehen:

```
DNS_REBINDOK_N='8'
DNS_REBINDOK_1_DOMAIN='rfc-ignorant.org'
DNS_REBINDOK_2_DOMAIN='spamhaus.org'
DNS_REBINDOK_3_DOMAIN='ix.dnsbl.manitu.net'
DNS_REBINDOK_4_DOMAIN='multi.surbl.org'
DNS_REBINDOK_5_DOMAIN='list.dnswl.org'
DNS_REBINDOK_6_DOMAIN='bb.barracudacentral.org'
DNS_REBINDOK_7_DOMAIN='dnsbl.sorbs.net'
DNS_REBINDOK_8_DOMAIN='nospam.login-solutions.de'
```

4.5.3. DHCP-Server

OPT_DHCP Mit `OPT_DHCP` kann man einstellen, ob der DHCP-Server aktiviert wird.

DHCP_TYPE (optional)

Mit dieser Variable legt man fest, ob man die interne DHCP-Funktion des dnsmasq benutzt, oder ob man auf den externen ISC-DHCPD zurückgreifen will. Im Falle des ISC-DHCPD entfällt der Support für DDNS.

DHCP_VERBOSE aktiviert zusätzliche DHCP-Ausgaben im log.

DHCP_LS_TIME_DYN legt die standard Lease-Time für dynamisch vergebene IP-Adressen fest.

DHCP_MAX_LS_TIME_DYN legt die maximale Lease-Time für dynamisch vergebene IP-Adressen fest.

DHCP_LS_TIME_FIX Standard Lease-Time für statisch zugeordnete IP-Adressen.

DHCP_MAX_LS_TIME_FIX legt die maximale Lease-Time für statisch zugeordnete IP-Adressen fest.

DHCP_LEASES_DIR legt das Verzeichnis für die Leases-Datei fest.

DHCP_LEASES_VOLATILE Befindet sich das Verzeichnis für die *Leases* in der Ram-Disk (da der Router z.B. von CD oder einem anderen nicht schreibbaren Medium bootet), gibt der Router beim Booten eine Warnung wegen einer fehlenden *Lease*-Datei aus. Diese Warnung entfällt, wenn man `DHCP_LEASES_VOLATILE` auf *yes* setzt.

DHCP_WINSSERVER_1 legt die Adresse des ersten WINS-Server fest. Bei installiertem und aktiviertem WINS-Server wird die Adresse von WINS-Server des SAMBA-Paketes übernommen.

DHCP_WINSSERVER_2 legt die Adresse des zweiten WINS-Server fest. Bei installiertem und aktiviertem WINS-Server wird die Adresse von WINS-Server des SAMBA-Paketes übernommen.

Lokale DHCP-Range

DHCP_RANGE_N Anzahl der DHCP-Ranges

DHCP_RANGE_x_NET Referenz zu einem in `IP_NET_x` definiertem Netz

DHCP_RANGE_x_START legt die erste zu vergebende IP-Adresse fest.

DHCP_RANGE_x_END legt die letzte zu vergebende IP-Adresse fest. Die beiden Variablen `DHCP_RANGE_x_START` und `DHCP_RANGE_x_END` kann man auch leer lassen, es wird dann keine DHCP-Range angelegt und nur die weiteren Variablen genutzt, um einem Host der per MAC-Zuordnung seine DHCP-IP bezieht, die Werte der Variablen zu übergeben.

DHCP_RANGE_x_DNS_SERVER1 legt die Adresse des DNS-Server für DHCP-Hosts des Netzes fest. Diese Variable ist optional. Wird hier nichts eingetragen, oder die Variable einfach weggelassen, wird die IP-Adresse, des zugeordneten Netzes verwendet. Es ist auch möglich, diese Variable auf 'none' zu setzen. Dann wird kein DNS-Server übertragen.

DHCP_RANGE_x_DNS_SERVER2 legt die IP-Adresse des zweiten DNS-Servers fest. Es gelten die gleiche Option wie in der vorherigen Variable

DHCP_RANGE_x_DNS_DOMAIN legt eine spezielle DNS-Domain für DHCP-Hosts dieser Range fest. Diese Variable ist optional. Wird hier nichts eingetragen, oder die Variable einfach weggelassen, wird der Default DNS-Domain `DOMAIN_NAME` verwendet.

DHCP_RANGE_x_NTP_SERVER legt die Adresse des NTP-Server für DHCP-Hosts dieser Range fest. Diese Variable ist optional. Wird hier nichts eingetragen, oder die Variable einfach weggelassen, wird die IP-Adresse, die in `DHCP_RANGE_x_NET` referenzierten Netzes, verwendet. Es ist auch möglich, diese Variable auf 'none' zu setzen. Dann wird kein NTP-Server übertragen.

DHCP_RANGE_x_GATEWAY legt die Adresse des Gateways für diese Range fest. Diese Variable ist optional. Wird hier nichts eingetragen, oder die Variable einfach weggelassen, wird die IP-Adresse, des in `DHCP_RANGE_x_NET` referenzierten Netzes, verwendet. Es ist auch möglich, diese Variable auf 'none' zu setzen. Dann wird kein Gatway übertragen.

DHCP_RANGE_x_MTU legt die MTU für Clients in dieser Range fest. Diese Variable ist optional.

DHCP_RANGE_x_OPTION_N gestattet die Angabe Nutzer-definierter Optionen für diesen Bereich. Die verfügbaren Optionen kann man dem Manual des dnsmasq entnehmen (`http://thekelleys.org.uk/dnsmasq/docs/dnsmasq.conf.example`). Sie werden ungeprüft übernommen, können also bei Fehlern zu Problemen mit dem DNS/DHCP-Server führen. Diese Variable ist optional.

Extra DHCP-Range

DHCP_EXTRA_RANGE_N legt die Anzahl von DHCP-Bereichen fest, die an nicht lokale Netze vergeben werden. Hierzu ist am Gateway zum entsprechenden Netz ein DHCP-Relay zu installieren.

DHCP_EXTRA_RANGE_x_START erste zu vergebende IP-Adresse.

DHCP_EXTRA_RANGE_x_END letzte zu vergebende IP-Adresse.

DHCP_EXTRA_RANGE_x_NETMASK Netzwerkmaske für diesen Bereich.

DHCP_EXTRA_RANGE_x_DNS_SERVER Adresse des DNS-Servers für diesen Bereich.

DHCP_EXTRA_RANGE_x_NTP_SERVER Adresse des NTP-Servers für diesen Bereich.

DHCP_EXTRA_RANGE_x_GATEWAY Adresse des Default-Gateway für diesen Bereich.

DHCP_EXTRA_RANGE_x_MTU legt die MTU für Clients in dieser Range fest. Diese Variable ist optional.

DHCP_EXTRA_RANGE_x_DEVICE Netzwerkinterface über den dieser Bereich erreicht wird.

Nicht zugelassene DHCP-Clients

DHCP_DENY_MAC_N Anzahl der MAC-Adressen von Host, dennen der Zugriff auf DHCP-Adressen verweigert wird.

DHCP_DENY_MAC_x MAC-Adresse des Hosts, dem der Zugriff auf DHCP-Adressen verweigert wird.

Unterstützung fürs Booten vom Netz

Der dnsmasq unterstützt Clients, die via Bootp/PXE übers Netz booten. Die dafür nötigen Informationen werden vom dnsmasq bereitgestellt und pro Subnetz und Host konfiguriert. Die dafür nötigen Variablen sind in den DHCP_RANGE_%- und HOST_%-Abschnitten untergebracht und beschreiben das zu bootende File (*_PXE_FILENAME), den Server, der dieses File bereitstellt (*_PXE_SERVERNAME und *_PXE_SERVERIP) und evtl. notwendige Optionen (*_PXE_OPTIONS). Weiterhin kann man den internen tftp-Server aktivieren, so daß das Booten komplett von dnsmasq unterstützt wird.

HOST_x_PXE_FILENAME DHCP_RANGE_x_PXE_FILENAME Hier wird das zu bootende Image angegeben. Im Falle von PXE wird hier der zu ladende pxe-Bootloader, wie z.B. pxegrub, pxelinux oder ein anderer passender Bootloader angegeben.

HOST_x_PXE_SERVERNAME HOST_x_PXE_SERVERIP DHCP_RANGE_x_PXE_SERVERNAME Name und IP des tftp-Servers, werden diese Variablen leer gelassen, wird der Router selbst als tftp-Server übermittelt.

DHCP_RANGE_x_PXE_OPTIONS HOST_x_PXE_OPTIONS Einige Bootloader benötigen spezielle Optionen zum Booten. So erfragt zum Beispiel pxegrub über die Option 150 den Namen der Menu-Datei. Diese Optionen können hier angegeben werden und werden dann ins Konfigfile übernommen. Im Falle von pxegrub könnte das z.B. wie folgt aussehen:

```
HOST_x_PXE_OPTIONS='150,"(nd)/grub-menu.lst"'
```

Sind mehrere Optionen nötig, werden sie einfach mit Leerzeichen voneinander getrennt angegeben.

4.5.4. DHCP-Relay

Das DHCP-Relay wird dann verwendet, wenn ein anderer DHCP-Server die Verwaltung der Ranges übernimmt, der nicht direkt von den Clients erreicht werden kann.

OPT_DHCPRELAY Dieser Wert ist auf 'yes' zu setzen, damit der Router als DHCP-Relay arbeitet. Es darf nicht gleichzeitig ein DHCP-Server aktiv sein.

Standard-Einstellung: `OPT_DHCPRELAY`='no'

DHCPRELAY_SERVER An dieser Stelle wird der richtige DHCP-Server eingetragen, an den die Anfragen weitergereicht werden sollen.

DHCPRELAY_IF_N DHCPRELAY_IF_x Mit `DHCPRELAY_IF_N` gibt man die Anzahl der Netzwerkkarten an, auf denen der Relay-Server lauschen soll. In `DHCPRELAY_IF_x` werden dann die entsprechenden Netzwerkkarten angegeben.

Das Interface, über das die Antworten des DHCP-Servers wieder reinkommen, muß in der Liste mit aufgeführt werden.Zusätzlich muss sichergestellt werden, dass die Routen auf dem Rechner, auf dem der DHCP-Server läuft, korrekt gesetzt sind. Die Antwort des DHCP-Servers geht an die IP des Interfaces, an dem der DHCP-Client hängt. Nehmen wir folgendes Scenario an:

- Relay mit zwei Interfaces
- Interfaces zum Client: eth0, 192.168.6.1
- Interfaces zum DHCP-Server: eth1, 192.168.7.1
- DHCP-Server: 192.168.7.2

Dann muss es auf dem DHCP-Server eine Route geben, über den die Antworten an die 192.168.6.1 ihr Ziel erreichen. Ist der Router, auf dem das Relay läuft, der default gateway für den DHCP-Server, ist bereits alles ok. Ist dem nicht so, wird eine extra Route benötigt. Ist der DHCP-Server ein Fli4l-Router, würde folgender Konfig-Eintrag dieses Ziel erreichen: IP_ROUTE_x='192.168.6.0/24 192.168.7.1'

Im Betrieb kann es zu Warnungen kommen, dass bestimmte Pakete ignoriert werden. Diese Warnungen kann man ignorieren, sie stören nicht den normalen Betrieb.

Beispiel:

```
OPT_DHCPRELAY='yes'
DHCPRELAY_SERVER='192.168.7.2'
DHCPRELAY_IF_N='2'
DHCPRELAY_IF_1='eth0'
DHCPRELAY_IF_2='eth1'
```

4.5.5. TFTP-Server

Der TFTP-Server wird dann verwendet, wenn der fli4l per TFTP Dateien ausliefern soll. Dies kann zum Beispiel dazu dienen, das ein Client per Netboot startet.

OPT_TFTP Aktiviert den internen TFTP-Server des dnsmasq. Standard-Wert ist 'no'.

TFTP_PATH Spezifiziert das Verzeichnis, in dem die Dateien liegen, die der tftp-Server an die Klienten ausliefern soll. Die entsprechenden Dateien sind mit Hilfe eines geeigneten Programms (z.B. scp) im entsprechenden Pfad abzulegen.

4.6. DSL - DSL über PPPoE, Fritz!DSL und PPTP

Fli4l unterstützt DSL in drei verschiedenen Varianten:

- PPPoE (externe, über Ethernet angeschlossene DSL-Modem, über die pppoe gefahren wird)
- PPTP (externe, über Ethernet angeschlossene Modem, über die pptp gefahren wird)
- Fritz!DSL (DSL über DSL-Adapter von AVM)

Man kann immer nur eine Variante auswählen, gleichzeitiger Betrieb ist leider noch nicht möglich.

Die Konfiguration dieser Varianten ähnelt sich, daher werden die allgemeinen Parameter vorab beschrieben und anschließend wird auf die Spezial-Optionen der einzelnen Varianten eingegangen. Der DSL-Zugang wird vom imond als Circuit[2] verwaltet, daher muß bei Aktivierung einer der DSL-Varianten auch der imond aktiviert werden (siehe START_IMOND (Seite 78)).

4.6.1. Allgemeine Konfigurationsvariablen

Die Pakete verwenden alle die gleichen Konfigurationsvariablen, sie unterscheiden sich nur durch den vorangestellten Paketnamen. Z.B. wird in allen Paketen der Nutzername verlangt, die Variable heißt lediglich je nach Paket PPPOE_USER, PPTP_USER oder FRITZDSL_USER. Im folgenden werden die Variablen ohne ihre Präfixe beschrieben, der fehlende Präfix wird durch einen Stern repräsentiert und in konkreten Beispielen wird von PPPOE ausgegangen (sie sind aber auch mit jedem anderen Präfix gültig).

***_NAME** Hier sollte ein Name für den Circuit vergeben werden - max. 15 Stellen lang. Dieser wird im imon-Client imonc angezeigt. Leerstellen (Blanks) sind nicht erlaubt.

Beispiel: PPPOE_NAME='DSL'

***_USEPEERDNS** Hiermit wird festgelegt, ob die vom Internet-Provider bei der Einwahl übergebenen Nameserver für die Dauer der Onlineverbindung in die Konfigurationsdatei des lokalen Nameservers eingetragen werden sollen.

Sinnvoll ist die Nutzung dieser Option also nur bei Circuits für Internet-Provider. Inzwischen unterstützen fast alle Provider diese Art der Übergabe.

Nachdem die Nameserver-IP-Adressen übertragen wurden, werden die unter DNS_FORWARDERS eingetragenen Nameserver durch die vom Provider vergebenen IP-Adressen als Forwarder verwendet. Danach wird der lokale Nameserver veranlasst, seine Konfiguration neu einzulesen. Dabei gehen bis dahin aufgelöste Namen nicht aus dem Nameserver-Cache verloren.

Diese Option bietet den Vorteil, immer mit den am nächsten liegenden Nameservern arbeiten zu können, sofern der Provider die korrekten IP-Adressen übermittelt - dadurch geht die Namensauflösung schneller.

Im Falle eines Ausfalls eines DNS-Servers beim Provider werden in der Regel die übergebenen DNS-Server-Adressen sehr schnell vom Provider korrigiert.

[2] Im Augenblick ist leider nur genau ein DSL-Circuit möglich – will man mehrere Circuits verwenden, muß man sich mehrere Floppies bauen

Trotz allem ist vor jeder ersten Einwahl die Angabe eines gültigen Nameservers in DNS_FORWARDERS zwingend erforderlich, da sonst die erste Anfrage nicht korrekt aufgelöst werden kann. Außerdem wird beim Beenden der Verbindung die originale Konfiguration des lokalen Nameservers wieder hergestellt.

Standard-Einstellung: *_USEPEERDNS='yes'

***_DEBUG** Soll pppd zusätzliche Debug-Informationen ausgeben, muss man *_DEBUG auf 'yes' setzen. In diesem Fall schreibt pppd zusätzlichen Informationen über die syslog-Schnittstelle.

WICHTIG: Damit diese auch über syslogd ausgegeben werden, muss die Variable OPT_SYSLOGD (s.o.) ebenso auf 'yes' gesetzt sein.

***_USER, *_PASS** Hier sind Benutzerkennung und Passwort für den jeweils benutzten Provider anzugeben. *_USER enthält die Benutzerkennung, *_PASS das Passwort.

WICHTIG: Für einen T-Online-Zugang ist folgendes zu beachten:

Der Username AAAAAAAAAAAATTTTTT#MMMM setzt sich aus der zwölfstelligen Anschlußkennung, der T-Online-Nummer und der Mitbenutzernummer zusammen. Hinter der T-Online-Nummer muß ein '#' angegeben werden, wenn die Länge der T-Online-Nummer kürzer als 12 Zeichen ist.

Sollte dies in Einzelfällen nicht zum Erfolg führen (offenbar abhängig von der Vermittlungsstelle), muß zusätzlich zwischen der Anschlußkennung und der T-Online-Nummer ein weiteres '#'-Zeichen eingefügt werden.

Ansonsten (T-Online-Nr ist 12stellig) sind keine '#'-Zeichen anzugeben.

Die Benutzerkennung muß bei T-Online mit '@t-online.de' abgeschlossen werden!

Beispiel:

```
PPPOE_USER='111111111111222222#0001@t-online.de'
```

Infos zu der Benutzerkennung bei anderen Providern finden sich in der FAQ:

- http://extern.fli4l.de/fli4l_faqengine/faq.php?list=category&catnr=3&prog=1

***_HUP_TIMEOUT** Hier kann die Zeit in Sekunden angegeben werden, nach welcher die Verbindung beendet werden soll, wenn nichts mehr über die DSL-Leitung läuft. Dabei steht ein Timeout von '0' oder 'never' für kein Timeout - bei 'never' wählt sich der Router zusätzlich nach einem Zwangsauflegen sofort wieder neu ein. Eine Veränderung des Dialmodes ist dann nicht mehr möglich - er muß auf 'auto' stehen und bleiben. 'never' ist momentan nur für PPPOE und FRITZDSL verfügbar.

***_CHARGEINT** Charge-Interval: Hier ist der Zeittakt in Sekunden anzugeben. Dieser wird dann für die Kosten-Berechnung verwendet.

Die meisten Provider rechnen minutengenau ab. In diesem Fall ist der Wert '60' richtig. Bei Providern mit sekundengenauer Abrechnung setzt man *_CHARGEINT besser auf '1'.

Leider wird bei DSL der Zeittakt nicht voll ausgenutzt, so wie es bei ISDN der Fall ist. Hier wird immer nach der Zeit, die in *_HUP_TIMEOUT angegeben ist, eingehängt.

Hier ist deshalb *_CHARGEINT lediglich für die Berechnung von Gebühren maßgeblich.

***_TIMES** Die hier angegebenen Zeiten bestimmen, wann dieser Circuit aktiviert werden soll und wann er wieviel kostet. Dadurch wird es möglich, zu verschiedenen Zeiten verschiedene Circuits mit Default-Routen zu verwenden (Least-Cost-Routing). Dabei kontrolliert der Daemon imond die Routen-Zuweisung.

Aufbau der Variablen:

```
PPPOE_TIMES='times-1-info [times-2-info] ...'
```

Jedes Feld times-?-info besteht aus 4 Unterfeldern - durch Doppelpunkt (':') getrennt.

1. Feld: W1-W2

   Wochentag-Zeitraum, z.B. Mo-Fr oder Sa-Su usw. Sowohl die deutsche als auch die englische Schreibweise ist erlaubt. Soll ein einzelner Wochentag eingetragen werden, ist zu W1-W1 schreiben, also z.B. Su-Su.

2. Feld: hh-hh

   Stunden-Bereich, z.B. 09-18 oder auch 18-09. 18-09 ist gleichbedeutend mit 18-24 plus 00-09. 00-24 meint den ganzen Tag.

3. Feld: Charge

   Hier werden in Euro-Werten die Kosten pro Minute angegeben, z.B. 0.032 für 3.2 Cent pro Minute. Diese werden unter Berücksichtigung der Taktzeit umgerechnet für die tatsächlich anfallenden Kosten, welche dann im imon-Client angezeigt werden.

   Feld: LC-Default-Route

   Der Inhalt kann Y oder N sein. Dabei bedeutet:

   Y: Der angegebene Zeitbereich wird beim LC-Routing als Default-Route verwendet.

   N: Der angegebene Zeitbereich dient nur zum Berechnen von Kosten, er wird beim automatischen LC-Routing jedoch nicht weiter verwendet.

Beispiel (als eine lange Zeile zu lesen):

```
PPPOE_TIMES='Mo-Fr:09-18:0.049:N
             Mo-Fr:18-09:0.044:Y
             Sa-Su:00-24:0.039:Y'
```

Wichtig: *Die bei *_TIMES angegebenen Zeiten muessen die ganze Woche abdecken. Ist das nicht der Fall, kann keine gültige Konfiguration erzeugt werden.*

Wenn die Zeitbereiche aller LC-Default-Route-Circuits ("Y") zusammengenommen nicht die komplette Woche beinhalten, gibt's zu diesen Lückenzeiten keine Default-Route. Damit ist dann Surfen im Internet zu diesen Zeiten ausgeschlossen!

Noch ein ganz einfaches Beispiel:

```
PPPOE_TIMES='Mo-Su:00-24:0.0:Y'
```

für diejenigen, die eine Flatrate nutzen.

Und noch eine letzte Bemerkung zum LC-Routing: *Feiertage werden wie Sonntage behandelt.*

***_FILTER** Fli4l legt automatisch auf, wenn während der über hangup timeout angegebenen Zeit keine Daten über das pppoe-Interface gehen. Leider wertet das Interface auch Datentransfers mit, die von außen kommen, z.B. durch Verbindungsversuche eines P2P-Clients wie eDonkey. Da man heutzutage eigentlich pemanent von anderen kontaktiert wird, kann es passieren, daß fli4l die DSL-Verbindung nie beendet.

Hier hilft die Option `*_FILTER`. Setzt man es auf yes, wird nur noch Verkehr gewertet, der von der eigenen Maschine generiert wird und externer Traffic wird komplett ignoriert. Da von draußen reinkommender Traffic in der Regel dazu führt, daß der Router oder dahinter liegende Rechner reagieren, indem sie z.B. Verbindungswünsche ablehnen, werden zusätzlich noch einige rausgehende Pakete ignoriert. Wie das genau funktioniert, kann man hier nachlesen:

- http://www.fli4l.de/hilfe/howtos/basteleien/hangup-problem-loesen/ und
- http://web.archive.org/web/20061107225118/http://www.linux-bayreuth.de/dcforum/DCForumID2/46.html.

Eine genauere Beschreibung des Ausdrucks und seiner Einbindung ist im Anhang zu finden - das ist aber nur interessant, wenn man Änderungen vornehmen will.

***_MTU *_MRU** Mit diesen optionalen Variablen können die sog. **MTU** (maximum transmission unit) und die **MRU** (maximum receive unit) eingestellt werden. Optional bedeutet, die Variable muß nicht in der Konfigurationsdatei stehen, sie ist bei Bedarf durch den Benutzer einzufügen!
Normal beträgt die MTU und die MRU jeweils 1492. Diese Einstellung sollte nur in Sonderfällen geändert werden! Diese Variablen gibt es nicht fuer OPT_PPTP.

***_NF_MSS** Bei manchen Providern treten Effekte folgender Art auf:

- Webbrowser bekommt eine Verbindung, macht aber danach nichts mehr,
- kleine Mail kann verschickt werden, große Mail nicht,
- ssh funktioniert, scp hängt nach dem initialen Verbindungsaufbau.

Um diese Probleme zu umgehen, manipuliert Fli4l standardmässig die MTU. In einige Faellen reicht das allerdings nicht, daher gestattet Fli4l explizit das Setzen der MSS (message segment size) auf einen vom Provider vorgegebenen Wert. Falls der Provider nichts vorgibt, ist 1412 ein guter Startwert zum probieren; falls er die MTU vorgibt, tragen sind hier 40 Byte weniger einzutragen ($mss = mtu - 40$). Diese Variable ist optional, was bedeutet, die Variable muß nicht in der Konfigurationsdatei stehen, sie ist bei Bedarf durch den Benutzer einzufügen! Diese Variable gibt es nicht fuer OPT_PPTP.

4.6.2. OPT_PPPOE - DSL über PPPoE

Für die Kommunikation über einen DSL-Anschluss ist in der Regel das PPPoE-Paket notwendig, weil die Provider keinen richtigen Router, sondern lediglich ein DSL-Modem zur Verfügung

stellen. Zwischen dem fli4l-Router und dem Modem wird das Protokoll PPP benutzt, jedoch hier speziell über das Ethernet.

Dabei können eine oder zwei Ethernet-Karten im fli4l-Router zum Einsatz kommen:

- Nur eine Karte mit IP für das LAN und PPP zum DSL-Modem
- Zwei Karten: eine für IP im LAN, die andere für PPP zum DSL-MODEM

Die bessere Wahl ist die Alternative mit den zwei Ethernet-Karten. Dann sind beide Protokolle - IP und PPPoE - sauber voneinander getrennt.

Aber die Methode mit einer Ethernet-Karte funktioniert ebenso. In diesem Fall wird das T-DSL-Modem einfach mit an den Netzwerk-Hub angeschlossen. Es muss dann aber eventuell mit leichten Einbußen bei der maximalen Übertragungsgeschwindigkeit gerechnet werden.

Ob man nun eine oder zwei Karten im fli4l-Router hat, in jedem Fall sollte die Netzwerkkarte zum DSL-Modem im 10-Mbit/s-Half-Duplex-Betrieb laufen, um Kommunikationsprobleme zwischen Modem und Netzwerkkarte auszuschließen. Alle neueren PCI- aber nur einige ISA-Netzwerkkarten lassen sich auf verschiedene Modi konfigurieren. Dazu ist es notwendig, sich eine DOS-Boot-Diskette zu erstellen und das Konfigurationstool der Karte dort mit abzuspeichern. Dann bootet man den fli4l-Router mit dieser Diskette, startet das Konfigurationsprogramm und stellt die Karte fest auf den oben genannten Modus ein. Das Konfigurationsprogramm für die Karte liegt beim Kauf auf der Treiberdiskette bei oder kann von der Webseite des Kartenherstellers heruntergeladen werden. Eventuell findet man es auch bei einer Recherche in der NIC-Datenbank:

- http://www.fli4l.de/hilfe/nic-db/

Wenn man zwei Karten benutzt, sollte man die erste Karte für das LAN und die zweite für die Verbindung zum DSL-Modem verwenden.

Dabei braucht nur die erste Karte mit IP-Adressen belegt werden.

Das heißt:

```
IP_NET_N='1'                # Nur *eine* Karte mit IP-Adresse!
IP_NET_1xxx='...'              # Die üblichen Parameter
```

Bei `PPPOE_ETH` gibt man 'eth1' für die zweite Ethernetkarte an und definiert *keine* `IP_NET_2-xxx`-Variablen.

OPT_PPPOE Aktiviert die Unterstützung für PPPoE. Standard-Einstellung: `OPT_PPPOE='no'`.

PPPOE_ETH Name des Ethernet-Interfaces

'eth0'	1. Ethernet-Karte
'eth1'	2. Ethernet-Karte
...	...

Standard-Einstellung: `PPPOE_ETH='eth1'`

PPPOE_TYPE *PPPOE* steht für die Übertragung von PPP-Paketen über Ethernet-Leitungen. D.h., die zu übertragenden Daten werden im ersten Schritt vom ppp-Daemon in ppp-Pakete und dann in einem zweiten Schritt für die Übertragung übers Ethernet nochmals in pppoe-Pakete verpackt, um dann ans DSL-Modem geschickt zu werden. Das

Wert	Beschreibung
async	Die Pakete werden durch den pppoe-Daemon erzeugt; die Kommunikation zwischen *pppd* und *pppoed* erfolgt asynchron.
sync	Die Pakete werden durch den pppoe-Daemon erzeugt; die Kommunikation zwischen *pppd* und *pppoed* erfolgt syncron. Das führt zu einer effizienteren Kommunikation und damit zu einer geringeren Prozessorlast.
in_kernel	Die ppp-Pakete werden direkt an den Linux-Kern gereicht, der daraus pppoe-Pakete macht. Dadurch entfällt die Kommunikation mit einem zweiten Daemon und damit eine Menge Kopieraufwand, was wiederum zu geringerer Prozessorlast führt.

Tabelle 4.2.: Arten der pppoe-Paketerzeugung

zweite Verpacken kann durch den pppoe-Daemon oder durch den Kern erfolgen. Mittels `PPPOE_TYPE` wird die Art und Weise der pppoe-Paketerzeugung definiert.

Jemand hat mal einen Vergleich der verschiedenen Varianten gemacht und kam auf einem Fujitsu Siemens PCD-H, P75 zu den in Tabelle 4.3 dargestellten Ergebnissen[3].

Fli4l	NIC	Bandbreite (down stream)	CPU-Auslastung
2.0.8	rtl8029 + rtl8139	310 kB/s	100%
2.0.8	2x 3Com Etherlink III	305 kB/s	100%
2.0.8	SMC + 3Com Etherlink III	300 kB/s	100%
2.1.7	SMC + 3Com Etherlink III	375 kB/s	40%

Tabelle 4.3.: Bandbreite und CPU-Auslastung bei pppoe

PPPOE_HUP_TIMEOUT Verwendet man als PPPoE-Typ in_kernel und als Dialmode auto, kann man als Timeout 'never' angeben. Der Router legt dann nicht mehr auf und wählt sich nach einer Zwangstrennung des Providers automatisch wieder ein. Nachträgliche Änderungen des Dialmodes ist nicht mehr möglich.

4.6.3. OPT_PPPOE_CIRC - Mehrere DSL-Circuits über PPPoE (Experimentell)

Will man mehrere DSL-Zugänge verwalten, kann man das über `OPT_PPPOE_CIRC` tun. Setzt man dieses Opt auf *yes*, kann man mehrere PPPOE-Circuits definieren. Die Anzahl wird über `PPPOE_CIRC_N` bestimmt, die restlichen Optionen entsprechen denen der `OPT_PPPOE`, es wird lediglich ein *CIRC_x* eingeschoben, z.B. `PPPOE_CIRC_x_NAME` statt `PPPOE_NAME`.

4.6.4. OPT_FRITZDSL - DSL per Fritz!Card DSL

Hier wird die Internetverbindung per Fritz!Card DSL aktiviert. Fuer die Internetverbindung wird die Fritz!Card DSL von AVM benutzt. Da die Treiber fuer diese Karten nicht der GPL

[3]Die Zahlen wurden einem Posting in spline.fli4l entnommen und nicht weiter geprüft. Der Artikel trug die Message ID <caf9fkala1@bla.spline.inf.fu-berlin.de>.

unterliegen ist es nicht moeglich diese mit dem DSL Paket zu liefern. Es ist daher unbedingt nötig diese Treiber vorher von http://www.fli4l.de/download/stabile-version/avm-treiber/ herunter zu laden und einfach in das fli4l Verzeichnis zu entpacken.
Da diese Treiber zu gross fuer eine Diskette sind, ist es ausserdem unbedingt noetig, fli4l auf Festplatte o.ae. zu installieren, wenn man diese Treiber verwenden moechte.

Die Circuit-Unterstützung für die Fritz!Card DSL wurde mit freundlicher Unterstützung von Stefan Uterhardt (E-Mail: zer0@onlinehome.de) realisiert.

OPT_FRITZDSL Aktiviert die Unterstützung für Fritz!DSL. Standard-Einstellung: OPT_FRITZDSL='no'.

FRITZDSL_TYPE Es gibt verschiedene Fritz!-Karten, ueber die eine DSL-Anbindung erfolgen kann. Die verwendete Karte wird ueber FRITZDSL_TYPE eingestellt, wobei die in Tabelle 4.4 aufgefuehrten Typen zur Verfuegung stehen.

Kartentyp	Verwendung
fcdsl	Fritz!Card DSL
fcdsl2	Fritz!Card DSLv2
fcdslsl	Fritz!Card DSL SL
fcdslusb	Fritz!Card DSL USB
fcdslslusb	Fritz!Card DSL SL USB
fcdslusb2	Fritz!Card DSL USBv2

Tabelle 4.4.: Fritz-Karten

Standard-Einstellung:

```
FRITZDSL_TYPE='fcdsl'
```

FRITZDSL_PROVIDER Mit dieser Option wird der Typ der Gegenstelle eingestellt. Moegliche Optionen sind:
U-R2, ECI, Siemens, Netcologne, oldArcor, Switzerland, Belgium, Austria1, Austria2, Austria3, Austria4

In Deutschland handelt es sich fast immer um UR-2. Siemens und ECI kommen nur bei sehr alten Anschluessen zum Einsatz.
Fuer Schweiz und Belgien sollten die Optionen selbsterklaerend sein und in Oesterreich heisst es ausprobieren.
Sollte jemand fuer Oesterreich eine bessere Beschriftung der Optionen haben moege er diese bitte mitteilen.

Standard-Einstellung:

```
FRITZDSL_PROVIDER='U-R2'
```

4.6.5. OPT_PPTP - DSL über PPTP in Österreich/die Niederlande (EXPERIMENTAL)

In Österreich (und anderen europäischen Ländern) wird statt PPPoE das PPTP-Protokoll verwendet. Auch hier wird eine separate Ethernet-Karte an ein PPTP-Modem angeschlossen.

Ab Version 2.0 ist der Zugang über PPTP als Circuit realisiert - mit freundlicher Unterstützung von Rudolf Hämmerle (E-Mail: `rudolf.haemmerle@aon.at`).

Bei PPTP werden zwei Karten verwendet. Hierbei sollte man die erste Karte für den Anschluß des LANs verwenden und die zweite für die Verbindung zum DSL-Modem.

Nur die erste Karte darf mit IP-Adressen belegt werden.

Das heißt:

```
IP_NET_N='1'                    # Nur *eine* Karte mit IP-Adresse!
IP_NET_1xxx='...'               # Die üblichen Parameter
```

Bei `PPTP_ETH` gibt man 'eth1' für die zweite Ethernetkarte an und definiert *keine* `IP_NET_2-xxx`-Variablen.

OPT_PPTP Aktiviert die Unterstützung für PPTP. Standard-Einstellung: `OPT_PPTP`='no'.

PPTP_ETH Name des Ethernet-Interfaces

'eth0'	1. Ethernet-Karte
'eth1'	2. Ethernet-Karte
...	...

Standard-Einstellung: `PPTP_ETH`='eth1'

PPTP_MODEM_TYPE Es gibt verschiedene PPTP-Modemtypen, über die eine pptp-Anbindung erfolgen kann. Das verwendete Modem wird über `PPTP_MODEM_TYPE` eingestellt, wobei die in Tabelle 4.5 aufgeführten Typen zur Verfügung stehen.

Modemtyp	Verwendung
bbaa	Östereich
bcaa	Östereich
xdsl	Östereich, Inode xDSL@home
mxstream	die Niederlande

Tabelle 4.5.: PPTP-Modemtypen

Standard-Einstellung:

```
PPTP_MODEM_TYPE='bcaa'
```

Inode xDSL@home

Implementiert wurde die Unterstützung von Inode xDSL@home in Anlehnung an das im Supportbereich von Inode beschriebene[4].

Es gibt momentan evtl. noch Probleme mit dem Erneuern der Lease für das Interface (die IP für das Interface wird über dhcp zugeteilt und muß regelmäßig neu angefordert werden) und das Auflegen und wieder Einwählen per imonc funktioniert noch nicht so richtig. Hier ist Hilfe in Form von Patches oder zur Verfügung stellung als Testkaninchen willkommen.

Bei xdsl gibt es zwei weitere Optionen für pptp:

[4] Siehe http://www6.inode.at/support/internetzugang/xdsl_home/konfiguration_ethernet_linux.html

PPTP_CLIENT_REORDER_TO Der pptp-client, der für xdsl genutzt wird, muß unter Umständen Pakete zwichenpuffern und umordnen. Normalerweise wartet er 0,3s auf ein ausstehendes Paket. Mit dieser Variabel kann man das Timeout zwischen 0.00 (gar nicht puffern) und 10.00 variieren. Die Zeiten müssen immer mit zwei Nachkommstellen angegeben werden.

PPTP_CLIENT_LOGLEVEL Hier kann angegeben werden, wieviel Debug-Ausgaben der pptp-client produziert. Möglich sind 0 (wenig), 1 (default) und 2 (viel).

4.6.6. OPT_POESTATUS - PPPoE-Status-Monitor auf fli4l-Console

Auch diesen PPPoE-Status-Monitor für DSL-Verbindungen hat Thorsten Pohlmann entwickelt.

Bei `OPT_POESTATUS`='yes' kann auf der 3. fli4l-Console jederzeit der DSL-Status eingesehen werden. Auf die 3. Console kann mit der Tastenkombination ALT-F3 gewechselt werden, zurück auf die 1. Console mit ALT-F1.

4.6.7. OPT_PFC - Packet-Filter-Compiler

Installiert den Packet-Filter-Compiler auf dem Router, siehe Anhang PPPD und Active Filter (Seite 357).

PFC benötigt man nur, wenn man am von fli4l verwendeten Filterausdruck etwas ändern will. Will man das nicht, kann man diese Option auf no lassen.

4.7. DYNDNS - Update für DynamicDNS-Dienste

Dieses Paket ist dafür gedacht, automatisch bei jeder Einwahl einen dynamischen Hostname zu updaten. Folgende Dienste werden unterstützt:

Anbieter	Homepage	Preis	Wert für DYNDNS_x_PROVIDER
`Hostnamen`			
Anmerkungen			
FreeDNS (afraid.org)	`http://freedns.afraid.org`	kostenlos	AFRAID
`Unfassbar viele. Bei der letzten Zählung waren es 1358. Sie reichen von doof (b-i-t-c-h.la) über geschmacklos (bin-laden.la) und langweilig (direct-support.com) bis witzig (lag4free.de), k3wl (3-1-3-3-7.la) und cool (pulsarproject.com). Zusätzlich gibts noch „private" Domains, bei denen man nicht einfach so eine Subdomain kriegt, sondern den Besitzer um Erlaubnis fragen muss. Davon sind's über tausend. Eine komplette Liste gibt es unter` `http://freedns.afraid.org/domain/registry/`			
Als Passwort ist hier der letzte Teil (hinter dem Fragezeichen) der URL anzugeben, die man auf der Homepage von Afraid.org abrufen kann (Einloggen ⇒„Dynamic DNS" ⇒Die URL, die sich hinter dem Link „Direct URL" versteckt). Alle anderen Angaben werden ignoriert.			
CJB.NET	`http://www.cjb.net/`	kostenlos	CJB
`*.cjb.net`			
-			

Companity	http://www.staticip.de/	kostenlos	COMPANITY
*.staticip.de, *.dynamic-nameserver.com			
-			

DHS International	http://www.dhs.org/	kostenlos	DHS
*.dhs.org			
-			

DNS2Go	http://dns2go.deerfield.com/	ab 19,95$ pro Jahr	DNS2GO
*.dns2go.com, *.27south.com, *.idlegames.com, *.mygamesite.net, *.d2g.com, *.101main.com, *.dns2gooffice.com, *.idleplay.net, *.audio-stream.net, *.myip.org, *.linux-dude.com, *.linux-site.net, *.nameyourspace.net, *.myftpsite.net, *.my-net-space.net, *.dynamic-site.net, *.linux-dude.net, *.dynodns.net, *.101main.net, *.i989.net, *.d2gwebsite.com, *.idlegames.net, *.dns2go.info, *.dns2go.biz, *.d2g.biz			
-			

DNS-O-Matic	http://www.dnsomatic.com	kostenlos	DNSOMATIC
-			
DNS-O-Matic bietet eine kostenlose und einfache Möglichkeit, eine dynamische IP-Änderungen für mehrere Dienste mit einem einzigen Update bekannt zu geben. Wenn man mehrere Dienste gleichzeitig updaten will, muss man äll.dnsomatic.comäls Hostnamen verwenden.			

DtDNS	http://www.dtdns.com/	5 Hostnamen kostenlos, für mehr einmalig 5$	DTDNS
*.3d-game.com, *.4irc.com, *.b0ne.com, *.bbsindex.com, *.chatnook.com, *.darktech.org, *.deaftone.com, *.dtdns.net, *.effers.com, *.etowns.net, *.etowns.org, *.flnet.org, *.gotgeeks.com, *.scieron.com, *.slyip.com, *.slyip.net, *.suroot.com			
-			

DynAccess	http://dynaccess.de/	ab 5 EUR pro Jahr	DYNACCESS
*.dyn-fli4l.de, *.dyn-fli4l.info, *.dyn-eisfair.de, *.dynaccess.de, *.dynaccess.com, *.dynaccess.net, *.dynaccess.org, *.dynaccess.info, *.dynaccess.biz, *.dynaccess.ws, *.dynxs.de, *.dynxs.com, *.dynxs.net, *.dynxs.org, *.dynxs.info, *.dynxs.biz, *.dynxs.ws, *.dyn-access.de, *.dyn-access.com, *.dyn-access.net, *.dyn-access.org, *.dyn-access.info, *.dyn-access.biz, *.dyn-access.ws			
DynAccess bietet im Rahmen der FLI4L-DynAccess-Kooperation für die Subdomains *.dyn-fli4l.de, *.dyn-fli4l.info und *.dyn-eisfair.de Sondertarife an. Informationen hierzu gibt es auf der Internet-Seite http://www.dyn-fli4l.de/ bzw. http://www.dyn-eisfair.de/.			

DynDNS.org	http://www.dyndns.com/	kostenlos	DYNDNS
*.ath.cx, *.dnsalias.com, *.dnsalias.net, *.dnsalias.org, *.dynalias.com, *.dynalias.net, *.dynalias.org, *.dyndns.biz, *.dyndns.info, *.dyndns.org, *.dyndns.tv, *.dyndns.ws, *.gotdns.com, *.gotdns.org, *.homedns.org, *.homeftp.net, *.homeftp.org, *.homeip.net, *.homelinux.com, *.homelinux.net, *.homelinux.org, *.homeunix.com, *.homeunix.net, *.homeunix.org, *.is-a-geek.com, *.is-a-geek.net, *.is-a-geek.org, *.isa-geek.com, *.isa-geek.net, *.isa-geek.org, *.kicks-ass.net, *.kicks-ass.org, *.merseine.nu, *.mine.nu, *.myphotos.cc, *.serveftp.net, *.serveftp.org, *.shacknet.nu, *.game-host.org, *.game-server.cc, *.blogdns.com, *.blogdns.net, *.blogdns.org			
-			

DynDNS.org (custom)	http://www.dyndns.com/services/dns/custom/	$24.95 pro Jahr und Domain	D
fast alle Top-Level Domains werden unterstützt			
Günstigere Preise bei vielen Domains			

DynDNS DK	http://dyndns.dk/	kostenlos	DYNDNSDK
*.dyndns.dk, *.kyed.com, *.lir.dk, *.yaboo.dk			
-			

dyndns:free	http://dyndnsfree.de/	kostenlos	DYNDNSFREE
*.xxlspeed.de, *.smtp-host.de, *.mx-host.de, *.dnsuser.de, *.b-24.de, *.nr67.de, *.mymx.ws, *.mxhost.ws, *.pcweb.ws, *.dnsspeed.info			
-			

eisfair.net	http://www.intersales.de/it-infrastruktur/dyneisfair.html	ab 20 EUR pro Jahr	DYN
*.eisfair.net, *.fli4l.net			
Mit der Benutzung dieses Dienstes unterstützt man die Arbeit der FLI4L- und eisfair-Entwickler.			

DyNS	http://www.dyns.cx/	kostenlos	DYNSCX
.dyns.cx,.dyns.net,*.ma.cx,*.metadns.cx			
-			

IN-Berlin e.V.	http://www.in-berlin.de/news.html	ab 5 Euro/Monat	INBERLIN
*.dyn-berlin.de, *.in-butter.de			
-			

KONTENT	http://www.kontent.de/	ab 5,50 EUR/Monat für eine Domain	KONTENT
*			
Bei diesem Provider kann eine komplette Domain per DynDNS auf den heimischen Rechner umgeleitet werden.			

Nerdcamp.net	http://nerdcamp.net/dynamic/dns.cgi	kostenlos	NERDCAMP
*.nerdcamp.net			
-			

No-IP.com	http://www.no-ip.com/	kostenlos	NOIP
*.3utlilities.com, *.bounceme.net, *.hopto.org, *.myftp.biz, *.myftp.org, *.myvnc.com, *.no-ip.biz, *.no-ip.com, *.no-ip.info, *.no-ip.org, *.redirectme.net, *.servebeer.com, *.servecouterstrike.com, *.serveftp.com, *.servegame.com, *.servehalflife.com, *.servehttp.com, *.servemp3.com, *.servepics.com, *.servequake.com, *.sytes.net, *.zapto.org			
-			

noxaDynDNS	http://www.noxa.de/	kostenlos	NOXA
*.h3c.de			
-			

OVH.DE	http://www.ovh.de/	im Webhostingpaket enthalten	OVHDE
abhängig von den registrierten Domains			
-			

Regfish.com	http://www.regfish.de/	ab 1,00 EUR/Monat für eine Domain	REGFISH
*			
Dieser Provider bietet komplette Domain-Pakete an, man kann dann eine Subdomain oder die ganze Domain dynamisch aktualisieren			

SelfHost.de	http://selfhost.de/cgi-bin/selfhost	kostenlos	SELFHOST
*.selfhost.tv			
-			

Strato	http://www.strato.de/	im Webhostingpaket enthalten	STRATO
abhängig von den registrierten Domains			
-			

T-Link.de	http://www.t-link.de/	kostenlos	TLINK
*.t-link.de			
Dieser Provider bietet seinen Kunden eine kostenlose Domain an.			

ZoneEdit.com	http://zoneedit.com/	kostenlos	ZONEEDIT
???			
-			

Ich versuche, diese Daten aktuell zu halten. Trotzdem übernehme ich keine Haftung für die Richtigkeit dieser Daten. Wer einen Fehler oder eine Änderung entdeckt, darf mir gerne eine Mail (E-Mail: `fli4l@portfolio16.de`) schicken.

Diese Liste ist komplett, andere Provider werden ohne Änderung nicht unterstützt. Wie man das Paket um eigene Anbieter erweitern kann, steht im Anhang.

Der dynamische Hostname wird automatisch bei jeder Einwahl ins Internet aktualisiert. Das Paket beinhaltet eine Sperre, die das mehrmalige aktualisieren der gleichen IP verhindert, da dies bei einigen DynDNS-Anbietern nicht gerne gesehen wird und im Extremfall zur Sperrung des Accounts führen kann.

Hinweis: Es kann einige Minuten dauern, bis die Änderung des dynamischen Hostnamens wirksam wird.

Bevor man mit der Einrichtung dieses Paketes beginnen kann, muss man sich bei einem der oben genannten Anbietern einen Account holen. Falls man das schon hat, kann man sofort loslegen. Hat man noch keinen Account, so kann man sich an obiger Tabelle orientieren, um einen Hostname zu finden, der den Ansprüchen genügt und den persönlichen Geschmack trifft.

Für die nun folgende Konfiguration benötigt man folgende Daten:

- Name des Anbieters
- Benutzername
- Passwort
- Der DynDNS-Hostname

Die benötigten Angaben können je nach Anbieter variieren, ich versuche eine möglichst konsistene Konfiguration zu bieten. Manchmal ist z.B. der Hostname gleich dem Benutzernamen, in so einem Fall werde ich versuchen, immer das Host-Feld zu benutzen und den Benutzernamen einfach ignorieren. Jetzt aber los:

OPT_DYNDNS Steht dieser Parameter auf "yes", wird `OPT_DYNDNS` aktiviert.

DYNDNS_SAVE_OUTPUT Wird dieser Parameter auf "yes" gestellt, wird das Ergebnis der DynDNS-Anfrage(n) in einer Datei gespeichert und kann über den Webserver[5] abgefragt werden.

DYNDNS_N Hat man bei mehreren DynDNS-Anbietern einen Account und will deswegen bei jeder Einwahl mehrere Namen updaten, so ist dieser Wert entsprechend anzupassen.

DYNDNS_x_PROVIDER Hier wird der Name des zu benutzenden Providers angegeben (siehe Tabelle weiter oben und Hinweis in der Config-Datei).

DYNDNS_x_USER Benutzername bei dem DynDNS-Anbieter. Häufig ist dies eine E-Mail-Adresse, ein selbstgewählter Name oder gleich dem DynDNS-Hostname.

DYNDNS_x_PASSWORD Hier ist das Passwort des DynDNS-Accounts anzugeben. Aufpassen, dass niemand anderes beim Editieren der Config-Datei zusieht!

DYNDNS_x_HOSTNAME Hier ist der *komplette* DynDNS-Hostname des Accounts einzutragen. Beispielsweise könnte hier folgendes stehen:

- `cool.nerdcamp.net`
- `user.dyndns.org`
- `fli4luser.fli4l.net`

DYNDNS_x_CIRCUIT Hier kann angegeben werden, bei welchen Circuits dieser Hostname aktualisiert wird. Die einzelnen Circuits werden mit Leerzeichen voneinander getrennt. Es kann z.B. erwünscht sein, den Hostnamen nur bei der DSL-Einwahl zu benutzen. Hier ein paar Beispiele:

```
DYNDNS_1_CIRCUIT='1 2 3'          # Nur ISDN: Circuits 1 bis 3
oder
DYNDNS_1_CIRCUIT='pppoe'          # Nur DSL: pppoe-Circuit
oder
DYNDNS_1_CIRCUIT='dhcp'           # Update bei DHCP-Providern
                                  # (opt_dhcp wird benötigt)
oder
DYNDNS_1_CIRCUIT='pppoe 1'        # DSL und ISDN
```

[5] OPT_HTTPD im Paket HTTPD (Seite 131) auf http://www.fli4l.de/download/stabile-version/

DYNDNS_x_RENEW Manche Provider erwarten, dass alle n Tage ein Update ausgeführt wird, auch wenn sich die IP nicht verändert hat. Diesen Intervall kann man hier angeben. Gibt man keinen Wert an, wird nach 29 Tagen ein Update durchgeführt.

Zu beachten ist hierbei, dass ein Update nur bei einer Einwahl angestoßen wird - also bei einer Einwahl über DSL oder ISDN oder einer Erneuerung einer Lease bei einem via DHCP konfigurierten Interface, wie man es bei einem Kabelmodem findet. Findet über längere Zeit keine Einwahl statt, muß man das Update auf andere Weise anstoßen.

DYNDNS_x_EXTIP Ist diese Variable auf 'yes' gesetzt, wird die beim Update verwendete IP über einen externen Dienst wie z.B. checkip.dyndns.org ermittelt. Das kann notwendig sein, wenn der Router selbst nicht derjenige ist, der die externe IP erhält. Dabei ist zu beachten, dass der Router in diesem Falle momentan eine Änderung der externen IP nicht mitbekommt, den dyndns-Namenseintrag also nicht zeitnahe aktualisieren kann.

DYNDNS_ALLOW_SSL Ist diese Variable auf 'yes' gesetzt, wird das Update wenn möglich über SSL (verschlüsselte Verbindung) durchgeführt.

DYNDNS_LOOKUP_NAMES Ein Update der IP sollte eigentlich nur erfolgen, wenn sich die IP geändert hat. Viele Fli4l-Router haben jedoch keinen permanenten Speicher, auf der die Information über die registierte IP gesichert werden kann, daher steht diese Information direkt nach dem Booten dort nicht zur Verfügung. Um trotzdem unnötige Updates zu vermeiden, kann Fli4l in dieser Situation (und nur in dieser Situation) beim Namensdienst nach der aktuell registrierten IP fragen. Die ermittelte IP wird dann zwischengespeichert und für jedes weitere Update genutzt.

Zu beachten ist dabei, dass nach einem Reboot das Update-Intervall neu beginnt, wenn Fli4l den Namensdienst zur Ermittlung der IP nutzt.

DYNDNS_DEBUG_PROVIDER Ist diese Variable auf 'yes' gesetzt, wird ein trace des Update-Vorgangs aufgezeichnet, so dass man im Nachhinein bei einem Problem prüfen kann, was schief gegangen ist. Default: `DYNDNS_DEBUG_PROVIDER`='no'

4.8. EASYCRON - Befehle zeitgesteuert ausführen

Dieses Paket wurde von Stefan Manske E-Mail: `fli4l@stephan.manske-net.de` zusammengestellt und vom fli4l-Team an 2.1 angepaßt.

4.8.1. Konfiguration

Mit `OPT_EASYCRON` kann man über das entsprechende config-file gesteuert zu bestimmten Zeiten Befehle ausführen lassen.

Dabei werden folgende Einträge benutzt:

OPT_EASYCRON mit `OPT_EASYCRON`='yes' wird das Paket aktiviert

Standard-Einstellung: `OPT_EASYCRON`='no'

EASYCRON_MAIL Da immer wieder Probleme auftraten, daß der crond unerwünschte Mails verschickt, kann man dies generell mit diesem Flag verhindern.

Standard-Einstellung: `EASYCRON_MAIL`='no'

EASYCRON_N Die Anzahl der verschiedenen Befehle, die von cron gestartet werden sollen.

EASYCRON_x_CUSTOM Wer sich mit den Einstellungen in der crontab auskennt, kann hier für jeden Eintrag eigene Einstellungen wie MAILTO, PATH, ... einstellen. Mehrere Einträge müssen durch \\ getrennt werden. Hier sollte man sich aber wirklich mit cron auskennen.

Standard-Einstellung: `EASYCRON_CUSTOM=”`

EASYCRON_x_COMMAND In `EASYCRON_x_COMMAND` wird der gewünschte Befehl eigetragen, wie z.B.

```
EASYCRON_1_COMMAND='echo 1 '>' /dev/console'
```

EASYCRON_x_TIME In `EASYCRON_x_TIME` wird die Ausführungszeit gemäß der üblichen cron-Syntax eingetragen.

4.8.2. Beispiele

- Der Computer wünscht uns “Ein gutes neues Jahr”

```
EASYCRON_1_COMMAND = 'echo Ein gutes neues Jahr! > /dev/console'
EASYCRON_1_TIME    = '0 0 31 12 *'
```

- xxx wird von Montag bis Freitag jeweils von 7-20 Uhr zu jeder vollen Stunde ausgeführt.

```
EASYCRON_1_COMMAND = 'xxx'
EASYCRON_1_TIME    = '0 7-20 0 * 1-5'
```

- Der Router beendet jede Nacht um 03:40 die Internet-Verbindung die per DSL aufgebaut ist baut sie nach 5sec Wartezeit wieder auf. Die folgenden Devicenamen sind möglich: pppoe, ippp[1-9], ppp[1-9].

```
EASYCRON_1_COMMAND = 'fli4lctrl hangup pppoe; sleep 5; fli4lctrl dial pppoe'
EASYCRON_1_TIME    = '40 3 * * *'
```

Weitere Informationen zur cron-Syntax finden Sie unter

- http://www.pro-linux.de/artikel/2/146/der-batchdaemon-cron.html
- http://de.linwiki.org/wiki/Linuxfibel_-_System-Administration_-_Zeit_und_Steuerung#Die_Datei_crontab
- http://web.archive.org/web/20021229004331/http://www.linux-magazin.de/Artikel/ausgabe/1998/08/Cron/cron.html
- http://web.archive.org/web/20070810063838/http://www.newbie-net.de/anleitung_cron.html

4.8.3. Voraussetzungen

- Fli4l in einer Version > 2.1.0
- für ältere Versionen bitte die entspechenden opt_easycron-Versionen aus der OPT-Datenbank verwenden

4.8.4. Installation

`OPT_EASYCRON` wird einfach wie jedes andere OPT im aktuelle fli4l-Verzeichnis entpackt.

4.9. HD - Unterstützung von Festplatten, Flash-Karten, USB-Sticks, ...

4.9.1. OPT_HDINSTALL - Installation auf Festplatte/CompactFlash

Normalerweise läuft fli4l nur von einer Diskette. Eine Festplatte oder ein CompactFlash ist daher eigentlich nicht notwendig. Sollen relativ große Softwarepakete (wie z.B. die Treiber für die DSL-Karten von AVM) für fli4l genutzt werden, kann es knapp auf der Diskette werden...

Für diesen Fall ist die Installation auf Festplatte oder auf einem CompactFlash-Modul gedacht. Ein CompactFlash verhält sich wie eine IDE-Festplatte, so dass es im fli4l-Rechner wie eine Festplatte behandelt werden kann. Da ein CompactFlash keine beweglichen Teile enthält, ist es nicht zu hören - ein wesentlicher Vorteil gegenüber Festplatten. Dem gegenüber steht der höhere Preis...

Neben Compact-Flash Modulen und anderen Flash-Modulen, die sich wie eine IDE-Festplatte verhalten, sind noch die vorwiegend in Industrie-Rechnern zu findenden DiskOnChip™-Module vom M-Systems für fli4l geeignet.

Nicht zur Benutzung mit fli4l geeignet sind USB-Sticks und normale Flash-Speicher, die nur mit einem speziellen Lesegerät angesprochen werden können (SD, MMC, SmartMedia).

Im Folgenden werden die notwendigen Schritte zur Installation auf einer Festplatte erklärt. Da die Festplatten-Variante gegenüber der reinen Diskettenlösung etwas schwieriger zu installieren ist, empfehle ich, zunächst einmal mit der einfachen Bootdiskette zu üben. Läuft der Router einwandfrei, kann man sich – falls nicht alle gewünschten Softwarepakete auf die Diskette passen – mal an die Festplatteninstallation wagen.

Der übliche Weg ist die Installation mit einer Boot-DISKETTE, aber falls der Router nicht über einen Diskettenanschluß verfügt, kann auch von LAN-Boot aus installiert werden. Das `OPT_HDINSTALL` bereitet nur noch die Festplatte vor und kopiert keine Dateien mehr von der Diskette auf die Festplatte. Dies wird ganz normal über ein Remote-Update per Imonc oder über scp gemacht. Ein weitere Weg ist die Installation und Einrichtung der Festplatte über eine BOOT-CD.

Eine Einführung in die verschiedenen Festplatten- Installationsvarianten A oder B befindet sich am Anfang der fli4l-Dokumentation (Seite 15). Bitte unbedingt vorher lesen!

HD-Installation von Diskette in sechs einfachen Schritten

1. lauffähige fli4l-Diskette mit dem Paket base sowie `OPT_HDINSTALL` und `OPT_HDDRV` erstellen. Zusätzlich muss diese Diskette ein Remote-Update ermöglichen! Es muss also entweder `OPT_SSHD` und `OPT_SCP` aktiv sein oder `START_IMOND` auf 'yes' stehen, wobei letzteres in Verbindung mit ISDN (ohne Treiber, also `ISDN_TYPE='0'`) nicht immer auf eine Diskette passt.

2. den Router mit dieser Diskette booten.

3. am Router einloggen und den Befehl “hdinstall.sh” ausführen.

4. wenn die Aufforderung dazu erscheint die Dateien syslinux.cfg, kernel, rootfs.img, opt.img und rc.cfg mittels Imonc oder scp auf den Router nach /boot kopieren. Ich empfehle dazu mit zwei fli4l-Verzeichnissen zu arbeiten, eines für die Setup-Diskette und ein zweites für die spätere HD-Version. Bei der HD-Version stellen Sie die Variable `BOOT_TYPE`='hd' ein und bei der Setup-Diskette auf 'fd'.

   Beim Remote-Update müssen natürlich die Dateien der HD-Version auf den Router übertragen werden!

5. Diskette entfernen, Router herunterfahren und neu starten (unter Verwendung von halt/reboot/poweroff). Der Router bootet jetzt von der Festplatte

6. bei Problemen den folgenden Abschnitt gut durchlesen.

HD-Installation ausführlich erklärt (inklusive Beispiele)

Zuerst muß eine Router-Diskette erstellt werden, bei der in der Datei config/hd.txt das `OPT_HDINSTALL` mit den Installationsscripten und das `OPT_HDDRV`, welches die Treiber für IDE- und SCSI-Festplatten enthält, richtig konfiguriert wurden. Bitte dazu auch den Abschnitt zu `OPT_HDDRV` gründlich durchlesen!

Die Variable `BOOT_TYPE` in der base.txt wird auf 'fd' eingestellt, es soll ja schliesslich eine Setup-DISKETTE erstellt werden. Die Variable `MOUNT_BOOT` in der base.txt muß auf 'rw' eingestellt werden, damit später ein neues Archiv opt.img über das Netzwerk aufgespielt werden kann.

Anschließend wird der Router von dieser Diskette gebootet. Durch Eingabe von “hdinstall.sh” an der fli4l-Console wird dann das Installationsprogramm gestartet. Nach Beantwortung von ein paar Fragen wird die Festplatte vorbereitet. Zuletzt erscheint eine Aufforderung, dass man die für den Router benötigten Dateien per Remote-Update aufspielen soll.

Dieses Remote-Update keinesfalls vergessen, der Router bootet sonst nicht von der Festplatte. Zum Neustarten des Routers nach dem Remote-Update unbedingt reboot/halt/poweroff verwenden, andernfalls können die beim Remote-Update vorgenommenen Änderungen verloren gehen.

Falls die Festplatte bei der Installation nicht automatisch gefunden wurde kann es passieren, dass der Router nicht von HD bootet. In diesem Fall muss in der syslinux.cfg der Eintrag boot=/dev/<bei der Installation angegebene Platte> an die Append-Zeile angehängt werden.

Das Installationsscript kann sowohl direkt am Router als auch über ssh von einem anderen PC aus gestartet werden. Im jedem Fall muss man sich vorher durch Eingabe des Passwortes am Router anmelden. Als ssh-Client für Windows-Rechner empfehlen wir die Freeware Putty.

Konfiguration der Setup-Diskette

Bereits hier muss die Netzwerkkonfiguration richtig eingestellt sein, damit man die Dateien für die Festplatte über das Netzwerk aufspielen kann. Für ein Remote-Update mittels Imonc wird bei der Setup-Diskette auch der IMOND und damit eine gültige ISDN/DSL-Konfiguration

BOOT_TYPE	entsprechend dem Bootmedium für die Installation einstellen, idR Diskette (fd)
MOUNT_BOOT='rw'	notwendig, um später ein neues opt.img über Netzwerk auf die Platte kopieren zu können
OPT_HDINSTALL='yes'	notwendig um das Setup-Script und die Tools zum Formatieren der Partitionen auf der Diskette zu haben
OPT_HDDRV='yes'	notwendig, weil ohne Treiber nicht auf die Festplatte zugegriffen werden kann.
IMOND='yes'	nach dem Vorbereiten der Festplatte werden die eigentlichen Dateien per remote Update übertragen. Dazu benötigt man entweder den imond, scp (OPT_SSHD='yes', OPT_SCP='yes') oder ein anderes Paket, das einen Filetransfer erlaubt.

Tabelle 4.6.: Beispiel für die Konfiguration der Setup-Diskette

benötigt. Alternativ dazu kann man die Dateien per scp auf den Router kopieren, dazu bitte OPT_SCP='yes' (befindet sich im Paket SSHD) einstellen. Alle nicht unbedingt nötigen Pakete bitte weglassen, also kein SAMBA_LPD, LCD, Portforwarding usw.

Falls die Installation mit der Fehlermeldung

```
*** ERROR: can't create new partition table, see docu ***
```

abbricht, können mehrere Fehlerquellen in Frage kommen:

- die Festplatte ist in Benutzung, evtl. durch einen abgebrochenen Installationsversuch. Einfach neu booten und noch einmal versuchen.
- es gibt Hardwareprobleme, mehr dazu bitte im Anhang nachlesen.

Im letzten Schritt kann man nun die endgültige Fassung der Konfigurationsdateien erstellen und alle gewünschten Pakete hinzufügen.

Beispiele für eine fertige Installation nach Typ A und Typ B:

Ein Beispiel für jede Konfiguration finden Sie in Tabelle 4.7.

BOOT_TYPE='hd'	notwendig, da sie ja jetzt von Festplatte starten
MOUNT_BOOT='rw|ro|no'	nach Wahl. Um später neue fli4l-Archive über Netzwerk auf die Platte kopieren zu können ist 'rw' nötig.
OPT_HDINSTALL='no'	nach der erfolgreichen Installation ist dieses Paket nicht mehr notwendig.
OPT_HDDRV='yes'	notwendig, weil ohne Treiber nicht auf die Festplatte zugegriffen werden kann.
OPT_MOUNT	nur aktivieren, falls eine Datenpartition oder eine Swap-Partition erstellt wurde

Tabelle 4.7.: Beispiel für eine Installation nach Typ A oder B

Grundsätzlich sollte man auf Flash-Medien keine Swap-Partition anlegen, weil deren Anzahl von Schreibzyklen begrenzt ist und damit ein früher Defekt des Mediums zu erwarten ist. Das

Erstellen einer Swap-Partition wird nur angeboten, falls weniger als 32MB RAM im Router stecken und die Installation NICHT auf ein Flash-Medium durchgeführt wird!

4.9.2. OPT_MOUNT - Mounten von Dateisystemen auf Festplatte etc.

OPT_MOUNT mountet die Datenpartition nach /data, eine Prüfung der Partition auf Fehler wird bei Bedarf automatisch durchgeführt. Ein evtl. vorhandenes CD-ROM wird nach /cdrom gemountet, falls eine CD eingelegt ist. Für die swap-Partition wird das OPT_MOUNT nicht mehr benötigt!

OPT_MOUNT funktioniert nur nach einer Neuinstallation des Routers auf HD, weil dabei eine Konfigurationsdatei auf der Boot-Partition erstellt wird. Wenn das `OPT_MOUNT` mit einem Remote-Update auf einen bereits installierten Router übertragen wurde, muss diese Konfigurationsdatei manuell erstellt werden.

Auch bei einem Boot von CD-ROM kann das OPT_MOUNT nicht genutzt werden. Die CD kann in diesem Fall mit MOUNT_BOOT='ro' gemountet werden.

Die Datei hd.cfg auf der DOS-Partition hat für einen Router nach Typ B mit Swap und Datenpartition den folgenden Inhalt:

```
hd_boot='hda1'
hd_opt='hda2'
hd_swap='hda3'
hd_data='hda4'
hd_boot_uuid='4A32-0C15'
hd_opt_uuid='c1e2bfa4-3841-4d25-ae0d-f8e40a84534d'
hd_swap_uuid='5f75874c-a82a-6294-c695-d301c3902844'
hd_data_uuid='278a5d12-651b-41ad-a8e7-97ccbc00e38f'
```

Nicht existierende Partitionen werden einfach weggelassen, bei einem Router Typ A ohne weitere Partitionen auf einer SCSI-Festplatte sieht das also so aus:

```
hd_boot='sda1'
hd_boot_uuid='4863-65EF'
```

4.9.3. OPT_EXTMOUNT - Mounten von Dateisystemen auf Festplatte etc.

OPT_EXTMOUNT mountet beliebige Datenpartition auf jeden beliebigen Mountpoint im Dateisystem. Damit ist es möglich von Hand erstellte Dateisysteme zu mounten und beispielsweise für einen Rsync-Server zur Verfügung zu stellen.

EXTMOUNT_N Die Anzahl der Datenpartitionen die extra gemountet werden sollen.

EXTMOUNT_x_VOLUMEID Device, Label oder UUID des Volumens, was gemountet werden soll. Wenn z.B. die IDE Festplatte mit der Partition 1 am sekundären IDE Controller gemountet werden soll muss hier `EXTMOUNT_x_VOLUMEID='sdc1'` stehen.

Mit dem Befehl 'blkid' kann man sich Device, Label und UUID aller verfügbaren Volumen anzeigen lassen.

EXTMOUNT_x_FILESYSTEM Das verwendete Dateisystem der Partition. fli4l unterstützt zur Zeit nur die Dateisysteme isofs, fat, vfat, ext2 und ext3. Beim Standardwert `EXTMOUNT_x_FILESYSTEM='auto'` wird versucht das verwendete Dateisystem automatisch festzustellen.

EXTMOUNT_x_MOUNTPOINT Der Pfad (Mountpoint) im Dateisystem in der das Device eingehängt wird. Der Pfad muss vorher nicht existieren, er wird automatisch erzeugt.

EXTMOUNT_x_OPTIONS Wenn spezielle mount Optionen an den mount Aufruf übergeben werden sollen können diese hier angegeben werden.

```
EXTMOUNT_1_VOLUMEID='hda2'        # device
EXTMOUNT_1_FILESYSTEM='ext3'      # filesystem
EXTMOUNT_1_MOUNTPOINT='/mnt/data' # mountpoint for device
EXTMOUNT_1_OPTIONS=''             # extra mount options passed via mount -o
```

4.9.4. OPT_HDSLEEP – automatisches Abschalten für IDE-Festplatten einstellen

Eine IDE-Festplatte kann sich automatisch abschalten, wenn eine bestimmte Zeit ohne Aktivität verstreicht. Damit benötigt die Platte kaum noch Strom und macht keine Geräusche mehr. Wenn ein Zugriff auf die Festplatte erfolgt, läuft sie automatisch wieder an.

Nicht alle Festplatten vertragen häufiges Wiederanlaufen. Daher sollte man die Zeit nicht zu klein wählen. Ältere IDE-Platten und SCSI-Laufwerke bieten diese Funktion erst gar nicht an.

Bei Flash-Medien sind diese Einstellungen nicht sinnvoll und auch nicht notwendig. SCSI-Festplatten werden ebenfalls nicht unterstützt.

HDSLEEP_TIMEOUT Diese Variable legt fest, nach welcher Zeit ohne Zugriff sich die Festplatte in den Power-Down-Modus gehen soll. Dann schaltet sich die Festplatte automatisch nach der Wartezeit aus und beim nächsten Zugriff wieder ein. Hierbei sind Wartezeiten in Minutenabständen von einer bis 20 Minuten möglich sowie in Abständen von 30 Minuten von einer Halben bis zu fünf Stunden. Eine Wartezeit von 21 oder 25 Minuten wird also z.B. auf 30 Minuten aufgerundet. Manche Festplatten ignorieren auch zu hohe Werte und stoppen dann schon nach einigen Minuten. Bitte unbedingt die korrekte Funktion durchtesten, da dies sehr von der jeweiligen Hardware abhängig ist!

```
HDTUNE_TIMEOUT='2'              # wait 2 minutes until power down
```

4.9.5. OPT_RECOVER – Notfalloption

Diese Variable legt fest, ob die entsprechenden Funktionen zur Erstellung einer Notfalloption verfügbar sind. Wenn die Option aktiviert ist, wird der Befehl "mkrecover.sh" mit auf den Router übertragen. Mit diesem kann dann an der Kommandokonsole durch einfachen Aufruf, die Notfallinstallation aktiviert werden. Beim installierten Paket "HTTPD" kann die Übertragung der aktuell laufenden Installation in eine Notfallinstallation im Menü Recover durchgeführt werden.

Um die Notfallinstallation zu Nutzen, ist bei nächsten Reboot im Bootmenü die Auswahl Recover auszuwählen.

```
OPT_RECOVER='yes'
```

4.9.6. OPT_HDDRV - Treiber für Festplattencontroller

Das Treiber-Paket basiert auf dem OPT_SCSI, welches Kai-Christian Arndt (E-Mail: kai-christian.arndt@gmx.de) für fli4l zusammengestellt hat.

Mit OPT_HDDRV='yes' können die benötigten Treiber aktiviert und installiert werden.

HDDRV_N Die Anzahl der Treiber, die geladen werden sollen, wird mit HDDRV_N eingestellt.

HDDRV_x Mit HDDRV_1 usw. werden die entsprechenden Treiber für die verwendeten Host-Adapter ausgewählt. Eine Liste der unterstützten Hostadapter ist in der Beispiel-Konfigurationsdatei enthalten.

HDDRV_x_OPTION Mit HDDRV_x_OPTION können Optionen übergeben werden, die einige Treiber zum laden benötigen. Dies kann z.B. eine IO-Adresse für einen SCSI-Controller sein. Bei den meisten Treibern kann diese Variable einfach leer gelassen werden.

Im Anhang (Seite 363) finden Sie eine Übersicht der Fehler, die bei Festplatten und CompactFlash am häufigsten auftreten.

Jetzt folgen einige Beispiele wie bestimmte Treiber zu laden sind.

Beispiel 1: Zugriff auf IDE-Festplatten oder Compact-Flash Medien.

```
OPT_HDDRV='yes'             # install Drivers for Harddisk: yes or no
HDDRV_N='1'                 # number of HD drivers
HDDRV_1='ide-hd'            # this is the ide-disk driver
HDDRV_1_OPTION=''           # no need for options yet
```

Beispiel 2: Zugriff auf IDE-Festplatte und ein IDE-CDROM zusätzlich

```
OPT_HDDRV='yes'             # install Drivers for Harddisk: yes or no
HDDRV_N='2'                 # number of HD drivers
HDDRV_1='ide-hd'            # this is the ide-disk driver
HDDRV_1_OPTION=''           # no need for options yet
HDDRV_2='ide-cd'            # IDE CDROM driver
HDDRV_2_OPTION=''           # no need for options yet
```

Beispiel 3: Zugriff auf SCSI-Festplatte an einem Adaptec 2940

```
OPT_HDDRV='yes'             # install Drivers for Harddisk: yes or no
HDDRV_N='1'                 # number of HD drivers
HDDRV_1='aic7xxx'           # various aic7xxx based Adaptec SCSI
HDDRV_1_OPTION=''           # no need for options yet
```

Beispiel 4: Eine SCSI-CD-ROM am selben Controller zu Beispiel 3

```
OPT_HDDRV='yes'             # install Drivers for Harddisk: yes or no
HDDRV_N='2'                 # number of HD drivers
HDDRV_1='aic7xxx'           # various aic7xxx based Adaptec SCSI
HDDRV_1_OPTION=''           # no need for options yet
HDDRV_2='scsi-cd'           # SCSI CDROM driver
HDDRV_2_OPTION=''           # enter cdrom driver option here, not used yet
```

Beispiel 5: Beschleunigter IDE-Zugriff beim PC-Engines ALIX

```
OPT_HDDRV='yes'             # install Drivers for Harddisk: yes or no
HDDRV_N='2'                 # number of HD drivers
HDDRV_1='amd74xx'           # AMD PCI IDE driver (e.g. ALIX)
HDDRV_1_OPTION=''           # no need for options yet
HDDRV_2='ide-hd'            # this is the ide-disk driver
HDDRV_2_OPTION=''           # no need for options yet
```

Beispiel 6: Boot vom USB-Stick

```
OPT_HDDRV='yes'             # install Drivers for Harddisk: yes or no
HDDRV_N='1'                 # number of HD drivers
HDDRV_1='usb-hd'            # Boot from USB Hard Disk or Stick
HDDRV_1_OPTION=''           # no need for options
```

Beispiel 7: Zugriff auf IDE-Festplatte, IDE-CDROM und dazu noch eine SCSI- Festplatte am Adaptec 2940 sowie ein SCSI-CDROM an einem Adaptec 1542. Dies ist allerdings keine geeignete Hardwareausstattung für einen Router, das Beispiel steht nur zur Abschreckung hier drin ;-)

```
OPT_HDDRV='yes'             # install Drivers for Harddisk: yes or no
HDDRV_N='5'                 # number of HD drivers
HDDRV_1='ide-hd'            # this is the ide-disk driver
HDDRV_1_OPTION=''           # no need for options yet
# Für die IDE-Festplatte

HDDRV_2='ide-cd'            # IDE CDROM driver
HDDRV_2_OPTION=''           # no need for options yet
# Für das IDE-CDROM

HDDRV_3='aic7xxx'           # various aic7xxx based Adaptec SCSI
HDDRV_3_OPTION=''           # no need for options yet
# Adaptec 2940, SCSI-HD-Unterstützung gibt es automatisch dazu.

HDDRV_4='aha1542'           # Adaptec AHA154x Controller
HDDRV_4_OPTION=''           # no need for options yet
# Adaptec 1542 für das SCSI-CDROM

HDDRV_5='scsi-cd'           # SCSI CDROM driver
HDDRV_5_OPTION=''           # enter cdrom driver option here, not used yet
# der Treiber für SCSI-CD-ROM's muß extra dazu geladen werden.
```

4.10. HTTPD - Status-Webserver

4.10.1. OPT_HTTPD - Mini-Webserver als Statusmonitor

Wer aus irgendeinem Grund keine Möglichkeit hat, den IMONC zu benutzen, weil er z.B. einen Mac benutzt, kann den Webserver benutzen, um den Status des FLI4L-Routers abzurufen oder zu ändern. Mit `OPT_HTTPD='yes'` kann man den Statusmonitor verwenden.

Um den Status abzurufen, muss man in seinen Browser eine der folgenden Adressen eingeben:

```
http://fli4l/
http://fli4l.domain.lan/
http://192.168.6.1/
```

Hat der fli4l-Router einen abweichenden Namen, muss dieser statt "fli4l" verwendet werden. Dies gilt auch für den Domain-Namen und die obige IP-Adresse. Hat man den Webserver auf einen anderen Port gelegt (per `HTTPD_PORT`), muss man diesen mit angeben:

```
http://fli4l:81/
```

Es wird seit der Version 2.1.12 eine Login-Seite angezeigt, die nicht passwortgeschützt ist. Die passwortgeschützten Seiten befinden sich im Unterverzeichnis admin, also beispielsweise:

```
http://fli4l.domain.lan/admin/
```

Der Webserver lässt sich über folgende Variablen anpassen:

HTTPD_GUI_LANG Hiermit wird die Sprache eingestellt, in der das Webinterface dargestellt werden soll. Wird hier 'auto' eingetragen, wird die Spracheinstellung der Variablen `LOCALE` (in der base.txt) verwendet.

HTTPD_GUI_SKIN Diese Variable beeinflusst das Aussehen der Oberfläche. Das Paket *httpd_skins* stellt mit den Werten *fixed* und *notfixed* zwei alternative Darstellungen zur Verfügung. Standardwert dieser Variable ist *default*.

HTTPD_LISTENIP Der Webserver bindet sich normalerweise an eine sogenannte Wildcard-Adresse, so daß er auf einem beliebigen Interface angesprochen werden kann. Soll er sich nur an eine IP-Adresse binden, so kann es mit diesem Parameter eingestellt werden. Dazu die IP-Adressse wie folgt eintragen: `IP_NET_x_IPADDR`. Normalerweise bleibt dieser Parameter leer, damit die Standardeinstellung (ansprechbar auf einer beliebigen IP) greift.

Dieser Parameter dient lediglich dazu, den httpd nur an eine IP zu binden, so dass sich andere Instanzen an die anderen IPs des Routers binden können. Er kann nicht ohne weiteres dazu genutzt werden, den Zugang einzelner Subnetze zum Webinterface des Routers zu sperren. Dazu benötigt man weiterhin die Hilfe des Paketfilters.

Es ist auch möglich, hier durch Leerzeichen getrennt mehrere IP anzugeben.

HTTPD_PORT Soll der Webserver auf einem anderen Port laufen als 80, so ist dieser Wert anzupassen. Das ist normalerweise nicht zu empfehlen, da dann z.B. mit http://fli4l:81/ auf den Webserver zugegriffen werden muß.

HTTPD_PORTFW Setzt man diese Variable auf 'yes', kann man über das Webinterface Änderungen an der Portweiterleitung vornehmen. Es können Regeln gelöscht und hinzugefügt werden, Änderungen werden sofort wirksam. Änderungen an den Regeln gelten nur für die Laufzeit des Routers. Wird der Router neu gestartet, sind die Änderungen weg.

Diese Variable hat einen Defaultwert von 'yes'.

HTTPD_ARPING Der Webserver stellt den Online-Zustand der mit `HOSTS_x` aufgelisteten Hosts dar. Dazu verwendet er den *"Arp-Cache"*, ein Zwischenspeicher der die Adressen der lokalen Hosts zwischenspeichert. Hat ein Rechner lange nicht mehr mit dem Router kommuniziert, verschwindet seine Adresse aus dem *"Arp-Cache"* und der Host scheint aus zu sein. Will man den *"Arp-Cache"* aktuell halten (also das rausfallen eigentlich nicht benötigter Einträge verhindern), kann man `HTTPD_ARPING` auf *'yes'* setzen.

4.10.2. Nutzerverwaltung

Der Webserver bietet eine ausgefeilte Benutzerverwaltung:

HTTPD_USER_N Hiermit wird die Anzahl der Benutzer eingestellt. Wird diese Variable auf 0 gesetzt, wird die Benutzerverwaltung komplett deaktiviert und jeder hat die Möglichkeit, auf den Webserver zuzugreifen.

HTTPD_USER_x_USERNAME HTTPD_USER_x_PASSWORD HTTPD_USER_x_RIGHTS Hier werden Benutzername und Passwort der einzelnen Benutzer eingetragen. Desweiteren wird für jeden Benutzer angegeben, auf welche Funktionen des Webservers er zugreifen darf. Diese Funktion wird mit der Variable `HTTPD_RIGHTS_x` geregelt. Im einfachsten Fall steht dort nur 'all', was bedeutet, dass der entsprechende Benutzer alles darf. Ansonsten hat die Variable den folgenden Aufbau:

```
'bereich1:recht1,recht2,... bereich2:...'
```

Statt für einen Bereich die einzelnen Rechte anzugeben, darf auch hier das Wort "all" eingesetzt werden, was wiederum heißt, das dieser Benutzer in diesem Bereich alle Rechte hat. Dabei gibt es folgende Bereiche und Rechte:

Bereich "status" Alles, was im Menü Status zu sehen ist.

- **view** Der Benutzer darf alle Menüpunkte aufrufen.
- **dial** Der Benutzer darf wählen und auflegen.
- **boot** Der Benutzer darf den Router herunterfahren & neu starten.
- **link** Der Benutzer darf Kanalbündlung an- und abschalten.
- **circuit** Der Benutzer darf den Circuit wechseln.
- **dialmode** Der Benutzer daf den Dialmode (Auto, Manual, Off) ändern.
- **conntrack** Der Benutzer darf die aktuell über den Router laufenden Verbindungen ansehen.
- **dyndns** Der Benutzer darf Log-Meldungen des `DYNDNS` (Seite 118) Paketes sehen.

Bereich "logs" Alles, was mit Logdateien zu tun hat (Verbindungen, Anrufe, Syslog)

- **view** Der Benutzer darf die Logdateien betrachten.
- **reset** Der Benutzer darf die Logdateien löschen.

Bereich "support" Alles, was nützlich ist, wenn man beispielsweise in der Newsgroup Hilfestellung sucht.

- **view** Der Benutzer darf die Links zur Doku, fli4l-Webseite, usw. abrufen

systeminfo Der Benutzer darf detaillierte Informationen zur Konfiguration und zum aktuellen Status des Routers (z. B.: Firewall) abfragen.

Hier noch einige Beispiele:

HTTPD_USER_1_RIGHTS='all' Diese Angabe erlaubt einem Benutzer alles!

HTTPD_USER_2_RIGHTS='status:view logs:view support:all' Dieser Benutzer darf sich zwar alles ansehen, aber nichts ändern.

HTTPD_USER_3_RIGHTS='status:view,dial,link' Dieser Benutzer darf sich den Status der Internetverbindung ansehen, wählen und Kanalbündelung ein- und ausschalten.

HTTPD_USER_4_RIGHTS='status:all' Dieser Benutzer darf alles mit der Internetverbindung machen und neu starten (natürlich auch herunterfahren). Er darf aber nicht die Logdateien sehen oder löschen, die Timetable darf er auch nicht sehen...

4.10.3. OPT_OAC - Online Access Control

OPT_OAC (optionale Variable)

Aktiviert das Modul 'Online Access Control'. Hierüber kann der Internet-Zugang jedes im Paket dns_dhcp (Seite 101) konfigurierten Rechners selektiv deaktiviert werden.

Es gibt auch ein Kommandozeilen-Tool, welches die Steuerung über andere Pakete wie z.B. EasyCron möglich macht:

/usr/local/bin/oac.sh

Die Optionen werden beim Aufruf angezeigt.

OAC_WANDEVICE (optionale Variable)

Schränkt die Online Zugangssperre auf Verbindungen über dieses Netzwerkdevice ein. z.B. 'pppoe'

OAC_INPUT (optionale Variable)

Bietet Schutz vor der Umgehung via Proxy.

OAC_INPUT='default' sperrt die konfigurierten Ports von: Privoxy, Squid, Tor, SS5, Transproxy.

OAC_INPUT='tcp:8080 tcp:3128' sperrt TCP Port 8080 und 3128. Dies ist eine Space-separierte Liste an zu sperrenden Ports mit dazugehörendem Protokoll (udp, tcp). Fehlt das Protokoll, werden udp und tcp Ports gesperrt.

Weglassen dieser Variable oder Inhalt 'no' deaktiviert die Funktion.

OAC_ALL_INVISIBLE (optionale Variable)

Schaltet die Gesamtübersicht aus, wenn mindestens eine sichtbare Gruppe existiert. Existiert nicht mindestens eine sichtbare Gruppe, so hat diese Variable keine Wirkung.

OAC_LIMITS (optionale Variable)

Gibt eine durch Leerzeichen getrennte Liste der Zeitlimits an, die zur Auswahl stehen. Die Limits werden in Minuten angegeben. Damit kann eine zeitlich limitierte Sperre oder Freigabe erreicht werden.

Default: '30 60 90 120 180 360 540'

OAC_MODE (optionale Variable)

Mögliche Werte: `'DROP'` oder `'REJECT'` (default)

OAC_GROUP_N (optionale Variable)

Anzahl der Clientgruppen. Dient der Übersichtlichkeit, erlaubt aber auch über das Web-Interface eine gesamte Gruppe gesammelt freizugeben oder zu sperren.

OAC_GROUP_x_NAME (optionale Variable)

Name der Gruppe - Dieser Name wird im Web-Interface angezeigt und ist auch über die das Kommandozeilenscript 'oac.sh' nutzbar.

OAC_GROUP_x_BOOTBLOCK (optionale Variable)

Wenn hier 'yes' steht, werden alle Clients der Gruppe beim Bootvorgang bereits gesperrt. Hilfreich, wenn Rechner idr. gesperrt sein sollen und nur ausnahmsweise nicht.

OAC_GROUP_x_INVISIBLE (optionale Variable)

Markiert die Gruppe als unsichtbar. Sinnvoll, wenn einige Rechner vorgesperrt werden sollen aber diese nicht als eigene Gruppe im Web-If sichtbar sein sollen. Das Kommandozeilentool oac.sh kann diese trotzdem ansprechen, wenn man das braucht, z.B. von easycron aus.

OAC_GROUP_x_CLIENT_N (optionale Variable)

Anzahl der Clients in der Gruppe.

OAC_GROUP_x_CLIENT_x (optionale Variable)

Name des Clients - Siehe Paket dns_dhcp (Seite 101).

OAC_BLOCK_UNKNOWN_IF_x (optionale Variable)

Liste der in base.txt definierten Interfaces, auf welchen nur in dns_dhcp.txt definierte Hosts ins Internet dürfen. Nicht-definierte Hosts sind hier damit generell gesperrt.

4.11. IPv6 - Internet Protokoll Version 6

4.11.1. Einleitung

Dieses Paket ermöglicht es, den fli4l-Router in vielerlei Hinsicht IPv6-tauglich zu machen. Dazu gehören Angaben über die IPv6-Adresse des Routers, die verwalteten IPv6-(Sub-)Netze, vordefinierte IPv6-Routen und Firewall-Regeln bzgl. IPv6-Paketen. Des Weiteren lassen sich IPv6-Dienste wie Multicasting und DHCPv6 konfigurieren. Schließlich ist es möglich, Tunnel zu IPv6-Anbietern automatisch aufbauen zu lassen. Dies funktioniert momentan aber leider nur mit so genannten 6in4-Tunneln, wie sie etwa der Anbieter "SixXS" unterstützt. Andere Technologien (AYIYA, 6to4, Teredo) werden zur Zeit nicht unterstützt.

IPv6 ist der Nachfolger des Internet-Protokolls IPv4. Hauptsächlich wurde es entwickelt, um die relativ kleine Menge von eindeutigen Internet-Adressen zu vergrößern: Während IPv4

ungefähr 2^{32} Adressen unterstützt,[6] sind es bei IPv6 bereits 2^{128} Adressen. Dadurch kann mit IPv6 jedem kommunizierenden Host eine eindeutige Adresse zugeordnet werden, und man ist nicht mehr auf Techniken wie NAT, PAT, Masquerading etc. angewiesen.

Neben diesem Aspekt spielten bei der Entwicklung des IPv6-Protokolls auch Themen wie Selbstkonfiguration und Sicherheit eine Rolle. Dies wird in späteren Abschnitten aufgegriffen.

Das größte Problem bei IPv6 ist dessen Verbreitung: Momentan wird IPv6 – verglichen mit IPv4 – nur sehr spärlich verwendet. Das liegt daran, dass IPv6 und IPv4 technisch nicht miteinander kompatibel sind und somit alle Software- und Hardware-Komponenten, die an der Paket-Weiterleitung im Internet beteiligt sind, für IPv6 nachgerüstet werden müssen. Auch bestimmte Dienste wie DNS (Domain Name System) müssen für IPv6 entsprechend aufgebohrt werden.

Hier tut sich also ein Teufelskreis auf: Die geringe Verbreitung von IPv6 bei Dienstanbietern im Internet führt zu Desinteresse seitens der Router-Hersteller, ihre Geräte mit IPv6-Funktionalität auszustatten, was wiederum dazu führt, dass Dienstanbieter die Umstellung auf IPv6 scheuen, weil sie fürchten, dass sich der Aufwand nicht lohnt. Erst langsam wendet sich das Blatt zugunsten von IPv6, nicht zuletzt unter dem immer stärkeren Druck der knappen Adressvorräte.[7]

4.11.2. Adressformat

Eine IPv6-Adresse besteht aus acht Zwei-Byte-Werten, die hexadezimal notiert werden:

Beispiel 1: `2001:db8:900:551:0:0:0:2`

Beispiel 2: `0:0:0:0:0:0:0:1` (IPv6-Loopback-Adresse)

Um die Adressen etwas übersichtlicher zu gestalten, werden aufeinander folgende Nullen zusammengelegt, indem sie entfernt werden und lediglich zwei unmittelbar aufeinander folgende Doppelpunkte verbleiben. Die obigen Adressen können also auch so geschrieben werden:

Beispiel 1 (kompakt): `2001:db8:900:551::2`

Beispiel 2 (kompakt): `::1`

Eine solche Kürzung ist aber nur höchstens einmal erlaubt, um Mehrdeutigkeiten zu vermeiden. Die Adresse `2001:0:0:1:2:0:0:3` kann also entweder zu `2001::1:2:0:0:3` oder zu `2001:0:0:1:2::3` verkürzt werden, nicht aber zu `2001::1:2::3`, da jetzt unklar wäre, wie die vier Nullen jeweils auf die zusammengezogenen Bereiche verteilt werden sollen.

Eine weitere Mehrdeutigkeit existiert, wenn eine IPv6-Adresse mit einem Port (TCP oder UDP) kombiniert werden soll: In diesem Fall darf man den Port nicht unmittelbar mit Doppelpunkt und Wert anschließen, weil der Doppelpunkt bereits innerhalb der Adresse verwendet wird und somit in manchen Fällen unklar wäre, ob die Port-Angabe nicht vielleicht doch eine Adress-Komponente darstellt. Deshalb muss in solchen Fällen die IPv6-Adresse in eckigen Klammern angegeben werden. Dies ist auch die Syntax, wie sie in URLs gefordert wird (etwa wenn im Web-Browser eine numerische IPv6-Adresse verwendet werden soll).

Beispiel 3: `[2001:db8:900:551::2]:1234`

Ohne die Verwendung von Klammern entsteht die Adresse `2001:db8:900:551::2:1234`, die unverkürzt der Adresse `2001:db8:900:551:0:0:2:1234` entspricht und somit keine Port-Angabe besitzt.

[6] nur ungefähr, weil einige Adressen speziellen Zwecken dienen, etwa für Broad- und Multicasting

[7] Inzwischen sind die letzten IPv4-Adressblöcke von der IANA vergeben worden.

4.11.3. Konfiguration

Allgemeine Einstellungen

Die allgemeinen Einstellungen beinhalten zum einen die Aktivierung der IPv6-Unterstützung und zum anderen die optionale Vergabe einer IPv6-Adresse an den Router.

OPT_IPV6 Aktiviert die IPv6 Unterstützung.

Standard-Einstellung: `OPT_IPV6='no'`

HOSTNAME_IP6 (optional) Diese Variable stellt explizit die IPv6-Adresse des Routers ein. Falls die Variable nicht gesetzt wird, wird die IPv6-Adresse auf die Adresse des ersten konfigurierten IPv6-Subnetzes gesetzt (`IPV6_NET_x`, siehe unten).

Beispiel: `HOSTNAME_IP6='IPV6_NET_1_IPADDR'`

Subnetz-Konfiguration

In diesem Abschnitt wird die Konfiguration von einem oder mehreren IPv6-Subnetzen vorgestellt. Ein IPv6-Subnetz ist ein IPv6-Adressraum, der über ein so genanntes Präfix spezifiziert wird und an eine bestimmte Netzwerk-Schnittstelle gebunden ist. Weitere Einstellungen betreffen das Veröffentlichen des Präfixes und des DNS-Dienstes innerhalb des Subnetzes sowie einen optionalen Router-Namen innerhalb dieses Subnetzes.

IPV6_NET_N Diese Variable enthält die Anzahl der verwendeten IPv6-Subnetze. Mindestens ein IPv6-Subnetz sollte definiert werden, um IPv6 im lokalen Netz nutzen zu können.

Standard-Einstellung: `IPV6_NET_N='0'`

IPV6_NET_x Diese Variable enthält für ein bestimmtes IPv6-Subnetz die IPv6-Adresse des Routers sowie die Größe der Netzmaske in CIDR-Notation. Soll das Subnetz öffentlich geroutet werden, so stammt es in der Regel vom Internet- bzw. Tunnel-Anbieter.

Wichtig: *Soll in dem Subnetz die zustandslose Selbstkonfiguration (siehe den Abschnitt zu `IPV6_NET_x_ADVERTISE` weiter unten) aktiviert werden, dann muss die Länge des Subnetz-Präfixes 64 Bit betragen!*

Beispiel: `IPV6_NET_1='2001:db8:1742::1/64'`

IPV6_NET_x_DEV Diese Variable enthält für ein bestimmtes IPv6-Subnetz den Namen der Netzwerk-Schnittstelle, an welches das IPv6-Netz gebunden wird. Dies kollidiert *nicht* mit den in der Basis-Konfiguration (`base.txt`) vergebenen Netzwerk-Schnittstellen, da einer Netzwerk-Schnittstelle sowohl IPv4- als auch IPv6-Adressen zugeordnet werden dürfen.

Beispiel: `IPV6_NET_1_DEV='eth0'`

IPV6_NET_x_ADVERTISE Diese Variable legt fest, ob das eingestellte Subnetz-Präfix im LAN mittels des radvd-Programms verbreitet wird. Dies wird für die sogenannte “stateless autoconfiguration” (zustandslose Selbstkonfiguration) verwendet und ist nicht mit DHCPv6 zu verwechseln. Mögliche Werte sind “yes” und “no”.

Es ist empfehlenswert, diese Einstellung zu aktivieren, es sei denn, alle Adressen im Netz werden statisch vergeben oder ein anderer Router ist bereits dafür zuständig, das Subnetz-Präfix anzukündigen.

Wichtig: *Die automatische Verteilung des Subnetzes funktioniert nur, wenn das Subnetz ein /64-Netz ist, d.h. wenn die Länge des Subnetz-Präfixes 64 Bit beträgt! Der Grund hierfür ist, dass die anderen Hosts im Netzwerk ihre IPv6-Adresse aus dem Präfix und ihrer Host-MAC-Adresse berechnen und dies nicht funktioniert, wenn der Host-Anteil nicht 64 Bit beträgt. Wenn die Selbstkonfiguration fehlschlägt, sollte also geprüft werden, ob das Subnetz-Präfix nicht vielleicht falsch angegeben worden ist (z.B. als /48).*

Beispiel: `IPV6_NET_1_ADVERTISE='yes'`

IPV6_NET_x_ADVERTISE_DNS Diese Variable legt fest, ob auch der lokale DNS-Dienst im IPv6-Subnetz angekündigt werden soll. Dies funktioniert nur, wenn die IPv6-Funktionalität des DNS-Dienstes via `DNS_SUPPORT_IPV6`='yes' aktiviert ist. Mögliche Werte sind "yes" und "no".

Standard-Einstellung: `IPV6_NET_1_ADVERTISE_DNS='no'`

IPV6_NET_x_DHCP Diese Variable aktiviert den DHCPv6-Dienst für dieses IPv6-Subnetz. Mögliche Werte sind "yes" und "no". DHCPv6 wird dabei nur verwendet, um Hosts in diesem Subnetz Informationen zum Domänen-Namen und zur Adresse des zu verwendenden DNS-Servers zukommen zu lassen. Die Zuordnung von IPv6-Adressen ist über DHCPv6 nicht möglich.

Die Adresse des DNS-Servers wird nur dann via DHCPv6 veröffentlicht, wenn die IPv6-Unterstützung des DNS-Dienstes via `DNS_SUPPORT_IPV6` im dns_dhcp-Paket eingeschaltet ist.

Wichtig: *Die Variablen `IPV6_NET_x_ADVERTISE_DNS` und `IPV6_NET_x_DHCP` schließen sich nicht gegenseitig aus, sondern können beide aktiviert sein. In diesem Fall kann die Adresse des DNS-Servers auf zweierlei Weise von Hosts im lokalen Netzwerk in Erfahrung gebracht werden.*

Pro Netzwerkschnittstelle darf höchstens ein gebundenes IPv6-Subnetz für DHCPv6 konfiguriert werden!

Standard-Einstellung: `IPV6_NET_1_DHCP='no'`

IPV6_NET_x_NAME (optional) Diese Variable legt einen Interface-spezifischen Hostnamen für den Router in diesem IPv6-Subnetz fest.

Beispiel: `IPV6_NET_1_NAME='fli4l-subnet1'`

Tunnel-Konfiguration

Dieser Abschnitt stellt die Konfiguration von 6in4-IPv6-Tunneln vor. Ein solcher Tunnel bietet sich an, wenn der eigene Internet-Anbieter kein IPv6 von Haus aus unterstützt. In diesem Fall wird mit einem bestimmten Internet-Host eines Tunnel-Brokers, dem so genannten PoP (Point of Presence), via IPv4 eine bidirektionale Verbindung aufgebaut, über die dann alle IPv6-Pakete

verpackt geroutet werden (deswegen 6 “in” 4, weil die IPv6-Pakete innerhalb von IPv4-Paketen gekapselt werden).[8] Damit das funktioniert, muss zum einen der Tunnel aufgebaut und zum anderen der Router so konfiguriert werden, dass die IPv6-Pakete, die ins Internet sollen, auch über den Tunnel geroutet werden. Der erste Teil wird in diesem Abschnitt konfiguriert, der zweite Teil wird im nächsten Abschnitt beschrieben.

IPV6_TUNNEL_N Diese Variable enthält die Anzahl der aufzubauenden 6in4-Tunnel.

Beispiel: `IPV6_TUNNEL_N='1'`

IPV6_TUNNEL_x_LOCALV4 Diese Variable enthält die lokale IPv4-Adresse des Tunnels oder den Wert 'dynamic', wenn die dynamisch zugewiesene IPv4-Adresse des aktiven WAN-Circuits verwendet werden soll. Letzteres ist nur sinnvoll, wenn es sich um einen Heartbeat-Tunnel handelt (siehe `IPV6_TUNNEL_x_TYPE` weiter unten).

Beispiele:

```
IPV6_TUNNEL_1_LOCALV4='172.16.0.2'
IPV6_TUNNEL_2_LOCALV4='dynamic'
```

IPV6_TUNNEL_x_REMOTEV4 Diese Variable enthält die entfernte IPv4-Adresse des Tunnels. Diese Angabe wird in der Regel vom Tunnel-Anbieter vorgegeben.

Beispiel (entspricht dem PoP deham01 von Easynet):

```
IPV6_TUNNEL_1_REMOTEV4='212.224.0.188'
```

Wichtig: *Wenn `PF_INPUT_ACCEPT_DEF` auf “no” steht, d.h. wenn die IPv4-Firewall manuell konfiguriert wird, dann wird eine Regel benötigt, die alle IPv6-in-IPv4-Pakete (IP-Protokoll 41) vom Tunnelendpunkt akzeptiert. Für den o.g. Tunnelendpunkt sähe die entsprechende Regel wie folgt aus:*

```
PF_INPUT_x='prot:41 212.224.0.188 ACCEPT'
```

IPV6_TUNNEL_x_LOCALV6 Diese Variable legt die lokale IPv6-Adresse des Tunnels inklusive verwendeter Netzmaske in CIDR-Notation fest. Diese Angabe wird vom Tunnelanbieter vorgegeben.

Wichtig: *Die* entfernte *IPv6-Adresse des Tunnels muss als Zielpunkt der IPv6-Default-Route konfiguriert werden (siehe `IPV6_ROUTE_x` weiter unten), damit IPv6-Pakete über den Tunnel und zurück geroutet werden können!*

Beispiel: `IPV6_TUNNEL_1_LOCALV6='2001:db8:1743::2/112'`

IPV6_TUNNEL_x_DEV (optional) Diese Variable enthält den Namen der zu erstellenden Tunnel-Netzwerkschnittstelle. Verschiedene Tunnel müssen unterschiedlich benannt werden, damit alles funktioniert. Falls die Variable nicht definiert ist, wird ein Tunnelname automatisch generiert (“6in4tun” + Tunnelindex).

Beispiel: `IPV6_TUNNEL_1_NAME='6in4'`

[8] Es handelt sich um das IPv4-Protokoll 41, “IPv6 encapsulation”.

IPV6_TUNNEL_x_MTU (optional) Diese Variable enthält die Größe der MTU (Maximum Transfer Unit) in Bytes, d.h. des größten Pakets, das noch getunnelt werden kann. Diese Angabe wird in der Regel vom Tunnelanbieter vorgegeben. Die Standard-Einstellung, falls nichts angegeben wird, lautet "1280" und sollte mit den meisten Tunneln verträglich sein.

Beispiel: `IPV6_TUNNEL_1_MTU='1280'`

IPV6_TUNNEL_x_TYPE Diese Variable bestimmt den Typ des Tunnels. Momentan werden die Werte "static" für statische Tunnel und "sixxs-heartbeat" für dynamische Heartbeat-Tunnel des Anbieters SixXS unterstützt. Wird die Variable leer gelassen, wird aus Kompatibilitätsgründen der Wert "sixxs-heartbeat" angenommen. Mehr zu Heartbeat-Tunneln steht im nächsten Absatz.

Beispiel: `IPV6_TUNNEL_1_TYPE='sixxs-heartbeat'`

Viele Tunnelanbieter verlangen, dass über den Tunnel permanent ein Lebenszeichen vom Router an den Anbieter gesandt wird, um zu verhindern, dass ein Host einen Tunnel in Anspruch nimmt, obwohl er ihn nicht nutzt. Dazu wird ein so genanntes Heartbeat-Protokoll (dt. "Herzschlag") verwendet. Zusätzlich verlangen Anbieter in der Regel eine erfolgreiche Anmeldung mit Benutzernamen und Passwort, um Missbrauch zu vermeiden. Soll ein solcher Heartbeat-Tunnel genutzt werden (wie sie etwa SixXS anbietet), dann müssen entsprechende Angaben gemacht werden, die im Folgenden beschrieben werden.

IPV6_TUNNEL_x_SIXXS_USERNAME Diese Variable enthält den Namen des Benutzers, der für den Tunnel-Login erforderlich ist.

Beispiel: `IPV6_TUNNEL_1_SIXXS_USERNAME='ABCDE-SIXXS'`

IPV6_TUNNEL_x_SIXXS_PASSWORD Diese Variable enthält das Passwort für den oben angegebenen Benutzernamen. Es darf keine Leerzeichen enthalten.

Beispiel: `IPV6_TUNNEL_1_SIXXS_PASSWORD='passwort'`

IPV6_TUNNEL_x_SIXXS_ID Diese Variable enthält den identifizierenden Namen des Tunnels. Der Name eines SixXS-Tunnels beginnt immer mit einem großen 'T'.

Beispiel: `IPV6_TUNNEL_1_SIXXS_ID='T1234'`

Routen-Konfiguration

Routen sind Wege für IPv6-Pakete. Damit der Router weiß, welches eingehende Paket er wohin schicken soll, greift er auf eine Routing-Tabelle zurück, in der genau diese Informationen zu finden sind. Im Falle von IPv6 ist es wichtig zu wissen, wohin IPv6-Pakete geschickt werden, die nicht ins lokale Netz sollen. Dafür muss eine Default-Route konfiguriert werden, die in der Regel alle Pakete an das andere Ende eines IPv6-Tunnels schickt. Andere Routen, die etwa benachbarte IPv6-Subnetze miteinander verbinden, sind natürlich ebenfalls erlaubt.

IPV6_ROUTE_N Diese Variable legt die Anzahl der zu spezifizierenden IPv6-Routen fest. Zumindest die Default-Route sollte existieren, damit der Router via IPv6 mit dem Internet kommunizieren kann. Somit sollte der Eintrag mindestens auf '1' gesetzt werden.

Standard-Einstellung: `IPV6_ROUTE_N='0'`

IPV6_ROUTE_x Diese Variable enthält die Route in der Form 'Zielnetz Gateway', wobei das Zielnetz in der CIDR-Notation erwartet wird. Für die Default-Route muss als Zielnetz `::/0` verwendet werden. Als Gateway ist der IPv6-Host am anderen Ende des Tunnels einzutragen, der vom Tunnel-Anbieter vorgegeben wird.

Beispiel: `IPV6_ROUTE_1='::/0␣2001:db8:1743::1'`

IPv6-Zusatzfunktionalität

Dieser Abschnitt konfiguriert weitere, optionale IPv6-Funktionalität.

IPV6_MULTICAST Diese Variable aktiviert die Unterstützung für IPv6-Multicasting. Mögliche Werte sind "yes" und "no".

Standard-Einstellung: `IPV6_MULTICAST='no'`

IPv6-Firewall

Wie für IPv4 wird auch für IPv6-Netzwerke eine Firewall benötigt, damit nicht jeder von außen jeden Rechner im lokalen Netz erreichen kann. Dies ist um so wichtiger, als dass jeder Rechner im Normalfall eine weltweit eindeutige IPv6-Adresse erhält, die dem Rechner permanent zugeordnet werden kann, da sie auf der MAC-Adresse der verwendeten Netzwerkkarte aufbaut.[9] Deshalb verbietet die Firewall erst einmal jegliche Zugriffe von außen und kann dann durch entsprechende Einträge in diesem Abschnitt Stück für Stück – je nach Bedarf – geöffnet werden.

Die Konfiguration der IPv6-Firewall entspricht im Großen und Ganzen der Konfiguration der IPv4-Firewall. Auf Besonderheiten und Unterschiede wird gesondert eingegangen.

PF6_INPUT_POLICY Diese Variable legt die Standard-Strategie für auf dem Router eingehende Pakete fest (INPUT-Chain). Mögliche Werte sind "REJECT" (Standard, weist alle Pakete ab), "DROP" (verwirft klammheimlich alle Pakete) und "ACCEPT" (akzeptiert alle Pakete). Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_INPUT_POLICY`.

Standard-Einstellung: `PF6_INPUT_POLICY='REJECT'`

PF6_INPUT_ACCEPT_DEF Diese Variable aktiviert die voreingestellten Regeln für den INPUT-Chain der IPv6-Firewall. Mögliche Werte sind "yes" und "no".

Die voreingestellten Regeln öffnen die Firewall für eingehende ICMPv6-Pings (ein Ping pro Sekunde als Limit) sowie für NDP-Pakete (Neighbour Discovery Procotol), das zur zustandslosen Selbstkonfiguration von IPv6-Netzen benötigt wird. Verbindungen von localhost sowie Antwortpakete zu lokal initiierten Verbindungen werden ebenfalls erlaubt. Schließlich wird die IPv4-Firewall dahingehend angepasst, dass für jeden Tunnel gekapselte IPv6-in-IPv4-Pakete vom Tunnelendpunkt akzeptiert werden.

Standard-Einstellung: `PF6_INPUT_ACCEPT_DEF='yes'`

[9] Eine Ausnahme existiert, wenn die so genannten "Privacy Extensions" aktiviert werden, weil dann ein Teil der IPv6-Adresse zufällig generiert wird. Dies wird jedoch momentan nicht unterstützt.

PF6_INPUT_LOG Diese Variable aktiviert das Logging aller zurückgewiesenen eingehenden Pakete. Mögliche Werte sind “yes” und “no”. Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_INPUT_LOG`.

Standard-Einstellung: `PF6_INPUT_LOG='no'`

PF6_INPUT_LOG_LIMIT Diese Variable konfiguriert das Log-Limit des INPUT-Chains der IPv6-Firewall, um die Log-Datei lesbar zu halten. Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_INPUT_LOG_LIMIT`.

Standard-Einstellung: `PF6_INPUT_LOG_LIMIT='3/minute:5'`

PF6_INPUT_REJ_LIMIT Diese Variable stellt das Limit für das Zurückweisen von eingehenden TCP-Paketen ein. Überschreitet ein solches Paket dieses Limit, wird das Paket klammheimlich verworfen (DROP). Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_INPUT_REJ_LIMIT`.

Standard-Einstellung: `PF6_INPUT_REJ_LIMIT='1/second:5'`

PF6_INPUT_UDP_REJ_LIMIT Diese Variable stellt das Limit für das Zurückweisen von eingehenden UDP-Paketen ein. Überschreitet ein solches UDP-Paket dieses Limit, wird es klammheimlich verworfen (DROP). Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_INPUT_UDP_REJ_LIMIT`.

Standard-Einstellung: `PF6_INPUT_UDP_REJ_LIMIT='1/second:5'`

PF6_INPUT_N Diese Variable enthält die Anzahl der IPv6-Firewallregeln für eingehende Pakete (INPUT-Chain). Standardmäßig werden zwei Regeln aktiviert: Die erste erlaubt allen lokalen Hosts Zugriff auf den Router über so genannte Link-Level-Adressen, und die zweite erlaubt die Kommunikation von Hosts aus dem ersten definierten IPv6-Subnetz mit dem Router.

Falls mehrere lokale IPv6-Subnetze definiert werden, muss die zweite Regel entsprechend oft vervielfältigt werden. Siehe hierzu die Konfigurationsdatei.

Standard-Einstellung: `PF6_INPUT_N='2'`

PF6_INPUT_x Diese Variable spezifiziert eine Regel für den INPUT-Chain der IPv6-Firewall. Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_INPUT_x`.

Unterschiede zur IPv4-Firewall:

- Anstatt `IP_NET_x` wird hier `IPV6_NET_x` benutzt.
- Anstatt `IP_ROUTE_x` wird hier `IPV6_ROUTE_x` benutzt.
- IPv6-Adressen müssen in eckigen Klammern eingeschlossen werden.

Beispiele:

```
PF6_INPUT_1='[fe80::0]/10 ACCEPT'
PF6_INPUT_2='IPV6_NET_1 ACCEPT'
PF6_INPUT_3='tmpl:samba DROP NOLOG'
```

PF6_INPUT_x_COMMENT Diese Variable enthält eine Beschreibung bzw. einen Kommentar zur zugehörigen INPUT-Regel.

Beispiel: `PF6_INPUT_3_COMMENT='no␣samba␣traffic␣allowed'`

PF6_FORWARD_POLICY Diese Variable legt die Standard-Strategie für von dem Router weiterzuleitenden Pakete fest (FORWARD-Chain). Mögliche Werte sind "REJECT" (Standard, weist alle Pakete ab), "DROP" (verwirft klammheimlich alle Pakete) und "ACCEPT" (akzeptiert alle Pakete). Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_FORWARD_POLICY`.

Standard-Einstellung: `PF6_FORWARD_POLICY='REJECT'`

PF6_FORWARD_ACCEPT_DEF Diese Variable aktiviert die voreingestellten Regeln für den FORWARD-Chain der IPv6-Firewall. Mögliche Werte sind "yes" und "no".

Die voreingestellten Regeln öffnen die Firewall für ausgehende ICMPv6-Pings (ein Ping pro Sekunde als Limit). Antwortpakete zu bereits erlaubten Verbindungen werden ebenfalls erlaubt.

Standard-Einstellung: `PF6_FORWARD_ACCEPT_DEF='yes'`

PF6_FORWARD_LOG Diese Variable aktiviert das Logging aller zurückgewiesenen weterzuleitenden Pakete. Mögliche Werte sind "yes" und "no". Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_FORWARD_LOG`.

Standard-Einstellung: `PF6_FORWARD_LOG='no'`

PF6_FORWARD_LOG_LIMIT Diese Variable konfiguriert das Log-Limit des FORWARD-Chain der IPv6-Firewall, um die Log-Datei lesbar zu halten. Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_FORWARD_LOG_LIMIT`.

Standard-Einstellung: `PF6_FORWARD_LOG_LIMIT='3/minute:5'`

PF6_FORWARD_REJ_LIMIT Diese Variable stellt das Limit für das Zurückweisen von weiterzuleitenden TCP-Paketen ein. Überschreitet ein solches TCP-Paket dieses Limit, wird es klammheimlich verworfen (DROP). Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_FORWARD_REJ_LIMIT`.

Standard-Einstellung: `PF6_FORWARD_REJ_LIMIT='1/second:5'`

PF6_FORWARD_UDP_REJ_LIMIT Diese Variable stellt das Limit für das Zurückweisen von weiterzuleitenden UDP-Paketen ein. Überschreitet ein solches UDP-Paket dieses Limit, wird es klammheimlich verworfen (DROP). Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_FORWARD_UDP_REJ_LIMIT`.

Standard-Einstellung: `PF6_FORWARD_UDP_REJ_LIMIT='1/second:5'`

PF6_FORWARD_N Diese Variable enthält die Anzahl der IPv6-Firewallregeln für weiterzuleitende Pakete (FORWARD-Chain). Standardmäßig werden zwei Regeln aktiviert: Die erste verhindert die Weiterleitung aller lokalen Samba-Pakete in nicht-lokale Netze, und die zweite erlaubt letzteres für alle anderen lokalen Pakete aus dem ersten definierten IPv6-Subnetz.

Falls mehrere lokale IPv6-Subnetze definiert werden, muss die letzte Regel entsprechend oft vervielfältigt werden. Siehe hierzu die Konfigurationsdatei.

Standard-Einstellung: `PF6_FORWARD_N='2'`

PF6_FORWARD_x Diese Variable spezifiziert eine Regel für den FORWARD-Chain der IPv6-Firewall. Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_FORWARD_x`.

Unterschiede zur IPv4-Firewall:

- Anstatt `IP_NET_x` wird hier `IPV6_NET_x` benutzt.
- Anstatt `IP_ROUTE_x` wird hier `IPV6_ROUTE_x` benutzt.
- IPv6-Adressen müssen in eckigen Klammern eingeschlossen werden.

Beispiele:

```
PF6_FORWARD_1='tmpl:samba DROP'
PF6_FORWARD_2='IPV6_NET_1 ACCEPT'
```

PF6_FORWARD_x_COMMENT Diese Variable enthält eine Beschreibung bzw. einen Kommentar zur zugehörigen FORWARD-Regel.

Beispiel: `PF6_FORWARD_1_COMMENT='no␣samba␣traffic␣allowed'`

PF6_USR_CHAIN_N Diese Variable enthält die Anzahl der vom Benutzer definierten IPv6-Firewall-Tabellen. Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_USR_CHAIN_N`.

Standard-Einstellung: `PF6_USR_CHAIN_N='0'`

PF6_USR_CHAIN_x_NAME Diese Variable enthält den Namen der entsprechenden benutzerdefinierten IPv6-Firewall-Tabelle. Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_USR_CHAIN_x_NAME`.

Beispiel: `PF6_USR_CHAIN_1_NAME='usr-myvpn'`

PF6_USR_CHAIN_x_RULE_N Diese Variable enthält die Anzahl der IPv6-Firewallregeln in der zugehörigen benutzerdefinierten IPv6-Firewall-Tabelle. Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_USR_CHAIN_x_RULE_N`.

Beispiel: `PF6_USR_CHAIN_1_RULE_N='0'`

PF6_USR_CHAIN_x_RULE_x Diese Variable spezifiziert eine Regel für die benutzerdefinierte IPv6-Firewall-Tabelle. Für eine genauere Beschreibung siehe die Dokumentation der Variable `PF_USR_CHAIN_x_RULE_x`.

Unterschiede zur IPv4-Firewall:

- Anstatt `IP_NET_x` wird hier `IPV6_NET_x` benutzt.
- Anstatt `IP_ROUTE_x` wird hier `IPV6_ROUTE_x` benutzt.
- IPv6-Adressen müssen in eckigen Klammern eingeschlossen werden.

PF6_USR_CHAIN_x_RULE_x_COMMENT Diese Variable enthält eine Beschreibung bzw. einen Kommentar zur zugehörigen Regel.

Beispiel: `PF6_USR_CHAIN_1_RULE_1_COMMENT='some␣user-defined␣rule'`

4.11.4. Web-GUI

Das Paket installiert einen Menüeintrag “Paketfilter (IPv6)” im Mini-HTTPD, unter dem die Einträge des für IPv6 konfigurierten Paketfilters eingesehen werden können.

4.12. ISDN - Kommunikation über aktive und passive ISDN-Karten

fli4l ist vornehmlich zum Einsatz als ISDN- und/oder DSL-Router gedacht. Mit der Einstellung `OPT_ISDN='yes'` wird das ISDN-Paket aktiviert. Voraussetzung ist eine ISDN-Karte, die von fli4l unterstützt wird.

Soll kein ISDN verwendet werden, kann mit der Einstellung `OPT_ISDN`='no' die ISDN-Installation abgeschaltet werden. Dann werden alle in diesem Kapitel folgenden ISDN-Variablen ignoriert.

Standard-Einstellung: `OPT_ISDN='no'`

4.12.1. Herstellen einer ISDN-Verbindung

Das Einwählverhalten von fli4l wird von drei verschiedenen Variablen bestimmt, `DIALMODE`, `ISDN_CIRC_X_ROUTE_X`, `ISDN_CIRC_X_TIMES`. Es wird von `DIALMODE` (Seite 80) (in <config>/base.txt) bestimmt, ob bei Eintreffen eines Paketes auf einem aktiven Circuit automatisch eine Verbindung aufgebaut werden soll oder nicht. `DIALMODE` kann folgende Werte annehmen:

auto Trifft ein Paket auf einem ISDN-Circuit (bzw. dem daraus abgeleiteten ISDN-Interface ippp*) ein, wird automatisch eine Verbindung aufgebaut. Ob und wann ein Paket auf einem ISDN-Circuit eintrifft, wird von `ISDN_CIRC_X_ROUTE_X` und `ISDN_CIRC_X_TIMES` bestimmt.

manual Im manuellen Modus muß der Verbindungsaufbau über imond/imonc angestoßen werden. Wie das geht, steht im Abschnitt über imonc/imond.

off Es werden keine ISDN-Verbindungen hergestellt.

Auf welchem der konfigurierten Circuits Pakete eintreffen und damit eine Einwahl auslösen können, wird über `ISDN_CIRC_X_ROUTE_X` definiert. Standardmäßig ist es auf '0.0.0.0/0', die sogenannte 'default route' gesetzt. Das heißt, daß alle Pakete, die das lokale Netz verlassen, über diesen Circuit gehen, wenn er aktiv ist. Ob und wann er aktiv ist, wird dabei von `ISDN_CIRC_X_TIMES` bestimmt, da fli4l über die definierten Circuits ein *least cost routing* durchführt (siehe Abschnitt Least-Cost-Routing - Funktionsweise (Seite 277) in der Dokumentation des Grundpaketes). Möchte man nicht alle Pakete über diesen Circuit leiten, sondern nur Pakete in ein bestimmtes Netz (z.B. Firmennetz), kann man hier ein oder mehrere Netze angeben. Dann richtet fli4l eine Route über das dem Circuit zugeordnete ISDN-Interface ein, die permanent aktiv ist. Wird nun ein Paket in dieses Netz geschickt, erfolgt automatisch der Verbindungsaufbau.

Wie schon erwähnt, beschreibt `ISDN_CIRC_X_TIMES` neben den Verbindungskosten eines Circuits auch, ob und wann ein Circuit mit einer 'default route' aktiv ist und damit einen Verbindungsaufbau auslösen kann. Das 'wann' spezifiert man über die Zeit, die ersten beiden Elemente einer time-info (z.B. Mo-Fr:09-18), das 'ob' durch den vierten Parameter lc-default-route (y/n).

Fli4l (bzw. imond) sorgt dann dafür, daß die Pakete, die das lokale Netz verlassen, immer über den zu diesem Zeitpunkt aktiven Circuit gehen und damit ein Herstellen der Verbindung zum Internet-Provider auslösen.

Zusammenfassend kann man also für die Standardanwendungsfälle folgendes sagen:

- Will man einfach nur ins Internet, stellt man `DIALMODE` auf auto, definiert 1-n Circuits, die als erste Route '0.0.0.0/0' haben und deren Zeiten (Zeiten mit lc-default-route = y) die gesamte Woche abdecken.

```
ISDN_CIRC_%_ROUTE_N='1'
ISDN_CIRC_%_ROUTE_1='0.0.0.0/0'
ISDN_CIRC_%_TIMES='Mo-Su:00-24:0.0148:Y'
```

- Will man nebenbei noch über eine spezielle Nummer ins Firmennetz, definiert man sich einen Circuit (oder mehrere Circuits) mit Route ungleich '0.0.0.0/0' und hat damit einen permanent aktiven Zugang zum Firmennetz.

```
ISDN_CIRC_%_ROUTE_N='1'
ISDN_CIRC_%_ROUTE_1='network/netmaskbits'
ISDN_CIRC_%_TIMES='Mo-Su:00-24:0.0148:Y'
```

4.12.2. ISDN-Karte

ISDN_TYPE ISDN_IO ISDN_IO0 ISDN_IO1 ISDN_MEM ISDN_IRQ Hier sind die technischen Daten für die ISDN-Karte anzugeben.

Die im Beispiel aufgeführten Werte funktionieren für eine TELES 16.3, wenn die Karte auf IO-Adresse 0xd80 eingestellt ist (über Dip-Switches). Bei einer anderen Einstellung der Karte muss der Wert geändert werden.

Häufig gemachter Fehler (Beispiel):

```
ISDN_IO='240' -- richtig wäre: ISDN_IO='0x240'
```

Bei IRQ 12 muss man einen eventuell vorhandenen PS/2-Maus-Anschluss im BIOS abschalten. Sonst besser einen anderen IRQ wählen! „Gute“ sind meist 5, 10 und 11.

`ISDN_TYPE` entspricht prinzipiell den Typen-Nummern für den HiSax-Treiber. Ausnahme: nicht-HiSax-Karten wie z.B. AVM-B1. Für diese wurde der Nummernkreis für die Typen erweitert (siehe unten). Die Liste der möglichen HiSax-Typen basiert auf `linux-2.x.y/Documentation/isdn/README.HiSax`.

Typ	Karte	Benötigte Parameter
Dummy Type-Nummer:		
0	no driver (dummy)	none
Typen-Nummern für HiSax-Treiber:		
1	Teles 16.0	irq, mem, io
2	Teles 8.0	irq, mem
3	Teles 16.3 (non PnP)	irq, io
4	Creatix/Teles PnP	irq, io0 (ISAC), io1 (HSCX)

Typ	Karte	Benötigte Parameter
5	AVM A1 (Fritz)	irq, io
5	AVM (Fritz!Card Classic)	irq, io
6	ELSA PCC/PCF cards	io or nothing for autodetect (the iobase is required only if you have more than one ELSA card in your PC)
7	ELSA Quickstep 1000	irq, io (from isapnp setup)
8	Teles 16.3 PCMCIA	irq, io
9	ITK ix1-micro Rev.2	irq, io (from isapnp setup?)
10	ELSA PCMCIA	irq, io (set with card manager)
11	Eicon.Diehl Diva ISA PnP	irq, io
11	Eicon.Diehl Diva PCI	no parameter
12	ASUS COM ISDNLink	irq, io (from isapnp setup)
13	HFC-2BS0 based cards	irq, io
14	Teles 16.3c PnP	irq, io
15	Sedlbauer Speed Card	irq, io
15	Sedlbauer PC/104	irq, io
15	Sedlbauer Speed PCI	no parameter
16	USR Sportster internal	irq, io
17	MIC card	irq, io
18	ELSA Quickstep 1000PCI	no parameter
19	Compaq ISDN S0 ISA card	irq, io0, io1, io (from isapnp setup io=IO2)
20	NETjet PCI card	no parameter
21	Teles PCI	no parameter
22	Sedlbauer Speed Star (PCMCIA)	irq, io (set with card manager)
23	reserved (AMD 7930)	n.a.
24	Dr. Neuhaus Niccy PnP	irq, io0, io1 (from isapnp setup)
24	Dr. Neuhaus Niccy PCI	no parameter
25	Teles S0Box	irq, io (of the used lpt port)
26	AVM A1 PCMCIA (Fritz!)	irq, io (set with card manager)
27	AVM PnP (Fritz!PnP)	irq, io (from isapnp setup)
27	AVM PCI (Fritz!PCI)	no parameter
28	Sedlbauer Speed Fax+	irq, io (from isapnp setup)
29	Siemens I-Surf 1.0	irq, io, memory (from isapnp setup)
30	ACER P10	irq, io (from isapnp setup)
31	HST Saphir	irq, io
32	Telekom A4T	none
33	Scitel Quadro	subcontroller (4*S0, subctrl 1...4)
34	Gazel ISDN cards (ISA)	irq,io
34	Gazel ISDN cards (PCI)	none
35	HFC 2BDS0 PCI	none
36	W6692 based PCI cards	none
37	2BDS0 S+, SP	irq,io
38	NETspider U PCI	none
39	2BDS0 SP/PCMCIA [10]	irq,io (set with card manager)
40	not used (hotplug)	n.a.
41	Formula-n enter:now PCI	none
81	ST5481 USB ISDN adapters	none
82	HFC USB based ISDN adapters	none
83	HFC-4S/8S based ISDN cards	none

[10] Bei älteren Versionen wurde Typ 84 mit Typ 39 angedeutet.

Typ	Karte	Benötigte Parameter
84	AVM Fritz!Card PCI/PCIv2/PnP	none
Typen-Nummern für Capi-Treiber:		
101	AVM-B1 PCI	no parameter
102	AVM-B1 ISA	irq, io
103	AVM-B1/M1/M2 PCMCIA	no parameter
104	AVM Fritz!DSL	no parameter
105	AVM Fritz!PCI	no parameter
106	AVM Fritz!PNP	irq, io (from isapnp setup)
107	AVM Fritz!Classic	irq, io
108	AVM Fritz!DSLv2	no parameter
109	AVM Fritz!USBv2	no parameter
110	AVM Fritz!DSL USB	no parameter
111	AVM Fritz!USB	no parameter
112	AVM Fritz!X USB	no parameter
113	AVM FRITZ!DSL USBv2	no parameter
Typen-Nummern für andere Treiber:		
201	ICN 2B	io, mem
Typen-Nummern für mISDN-Treiber (experimental):		
301	HFC-4S/8S/E1 multiport cards	no parameter
302	HFC-PCI based cards	no parameter
303	HFCS-USB Adapters	no parameter
304	AVM FritZ!Card PCI (v1 and v2) cards	no parameter
305	cards based on Infineon (former Siemens) chips: - Dialogic Diva 2.0 - Dialogic Diva 2.0U - Dialogic Diva 2.01 - Dialogic Diva 2.02 - Sedlbauer Speedwin - HST Saphir3 - Develo (former ELSA) Microlink PCI (Quickstep 1000) - Develo (former ELSA) Quickstep 3000 - Berkom Scitel BRIX Quadro - Dr.Neuhaus (Sagem) Niccy	no parameter
306	NetJet TJ 300 and TJ320 cards	no parameter
307	Sedlbauer Speedfax+ cards	no parameter
308	Winbond 6692 based cards	no parameter

Meine Karte ist eine Teles 16.3 NON-PNP ISA, also ist Type=3.

Für eine ICN-2B-Karte müssen IO und MEM gesetzt werden, zum Beispiel `ISDN_IO='0x320'`, `ISDN_MEM='0xd0000'`.

Bei neueren Teles-PCI-Karten muss type=20 (statt 21) verwendet werden. Die Dinger melden sich bei “cat /proc/pci” mit “tiger” oder so ähnlich. Sonst kann ich nichts zu diesen Werten beitragen, sorry.

Um die ISDN-Typen 104 bis 113 verwenden zu können, ist es vorher nötig, die passenden Treiberdateien von http://www.fli4l.de/download/stabile-version/avm-treiber/ herunterzuladen und in das fli4l-Verzeichnis zu entpacken. Da diese Treiber nicht der GPL

unterliegen, können sie leider nicht mit dem ISDN Paket mitgeliefert werden.
Weiter sind diese Treiber zu groß für eine Diskette, deshalb ist es unbedingt nötig, fli4l auf Festplatte o.ä. zu installieren, wenn man diese Treiber verwenden möchte.

Für die ISDN-Typen 81, 82, 109 bis 113 und 303 ist es nötig USB Support zu installieren und zu aktivieren. Siehe dazu USB - Support für USB-Geräte (Seite 247).

Um die ISDN-Typen 10, 22, 26, 39 oder 103 verwenden zu können ist es nötig PCMCIA PC-Card Unterstützung zu installieren und zu aktivieren. Siehe dazu PCMCIA - PC-Card Unterstützung (Seite 210).

Tips zu den Typen-Nummern bekommt man auch über die i4l-FAQ oder Mailingliste, wenn man wirklich nicht weiß, was da für eine Karte im PC steckt.

Die Kartentypen, die mit „from isapnp setup“ gekennzeichnet sind, müssen mit dem PnP-Tool isapnp initialisiert werden - wenn es sich tatsächlich um eine PnP-Karte handelt. Siehe dazu die Beschreibung im Kapitel OPT_PNP - Installation von isapnp tools (Seite 84).

Der ISDN-Typ 0 wird dann benötigt, wenn man das ISDN-Paket installieren will ohne ISDN-Karte; z.B. um imond verwenden zu können bei einem Netzwerkrouter.

ISDN_DEBUG_LEVEL Dieses gibt den Debug-Level für den HiSaX-Treiber an. Der Debug-Level setzt sich dabei durch Addition der folgenden Werte zusammen (Zitat aus der Orginal-Doku):

Number	Debug-Information
1	Link-level <–> hardware-level communication
2	Top state machine
4	D-Channel Q.931 (call control messages)
8	D-Channel Q.921
16	B-Channel X.75
32	D-Channel l2
64	B-Channel l2
128	D-Channel link state debugging
256	B-Channel link state debugging
512	TEI debug
1024	LOCK debug in callc.c
2048	More debug in callc.c (not for normal use)

Die Standardeinstellung (`ISDN_DEBUG_LEVEL`=’31’) sollte den meisten reichen.

ISDN_VERBOSE_LEVEL Hiermit kann man die “Geschwätzigkeit” des ISDN-Subsystems im fli4l-Kernel einstellen. Jeder Verbose-Level schließt die Level mit niedrigerer Nummer mit ein. Die Verbose-Level sind:

’0’	keine zusätzlichen Informationen
’1’	Es wird protokolliert, was eine ISDN-Verbindung ausgelöst hat
’2’ und ’3’	Anrufe werden protokolliert
’4’ und mehr	Die Datentransferrate wird regelmäßig protokolliert.

Die Meldungen werden über das Kernel-Logging-Interface ausgegeben, erscheinen also bei aktiviertem OPT_SYSLOGD (Seite 81) dort.

Wichtig: *Sollen Anrufe mit telmond protokolliert werden, diesen Wert nicht kleiner als 2 einstellen, da telmond sonst die Informationen fehlen, um Anrufe zu protokollieren.*

Standardeinstellung: ISDN_VERBOSE_LEVEL='2'

ISDN_FILTER Aktiviert den Filtermechanismus des Kerns, um ein ordnungsgemäßes Auflegen nach dem angegebenen Hangup-Timeout zu gewährleisten. Siehe http://www.fli4l.de/hilfe/howtos/basteleien/hangup-problem-loesen/ für genauere Informationen.

4.12.3. OPT_ISDN_COMP (EXPERIMENTAL)

Mit OPT_ISDN_COMP='yes' werden die LZS- und die BSD-Kompression aktiviert. Das Kompressionspaket hat freundlicherweise Arwin Vosselman (E-Mail: arwin(at)xs4all(dot)nl) zusammengestellt. Dieses Zusatzpaket hat Experimental-Status.

Standard-Einstellung: OPT_ISDN_COMP='no'

Hier im Einzelnen die erforderlichen Parameter für LZS-Kompression:

ISDN_LZS_DEBUG (EXPERIMENTAL) Debug-Level-Einstellung:

- '0' keine Debugging Information
- '1' normale Debugging Information
- '2' erweiterte Debugging Information
- '3' schwere Debugging Information (inkl. Dumping der Datenpakete)

Standard-Einstellung: ISDN_LZS_DEBUG='1'

Wer bei Problemen mit der Komprimierung noch mehr Debugmeldungen sehen möchte, setzt diese Variable auf '2'.

ISDN_LZS_COMP (EXPERIMENTAL) Stärke der Kompression (nicht Dekompression!). Bitte erst einmal auf dem Wert '8' stehen lassen. Werte von 0 bis 9 sind möglich.

Grössere Zahlen geben bessere Kompression, 9 erzeugt aber übermässige CPU-Last.

Standard-Einstellung: ISDN_LZS_COMP='8'

ISDN_LZS_TWEAK (EXPERIMENTAL) Auch diese Variable erst einmal auf '7' stehen lassen.

Standard-Einstellung: ISDN_LZS_TWEAK='7'

Außer diesen 3 Werten muss noch die Variable ISDN_CIRC_x_FRAMECOMP angepasst werden, s. nächstes Kapitel.

4.12.4. ISDN-Circuits

In der fli4l-Konfiguration können mehrere Verbindungen über ISDN definiert werden. Davon sind maximal 2 Verbindungen auch zur gleichen Zeit über eine ISDN-Karte möglich.

Die Definition solcher Verbindungen geschieht über sogenannte Circuits. Dabei wird pro Verbindung ein Circuit verwendet.

In der Beispiel-Datei config.txt sind zwei solcher Circuits definiert:

- Circuit 1: Dialout über Internet-By-Call-Provider Microsoft Network, Sync-PPP
- Circuit 2: Dialin/Dialout zu einem ISDN-Router (das könnte beispielsweise auch fli4l sein) über Raw-IP, z.B. als Zugang zum Firmen-Netz von irgendwo. Bei mir konkret ist das eine Linux-Box mit isdn4linux als "Gegner".

Soll der fli4l-Router lediglich als Internet-Gateway dienen, ist nur ein Circuit notwendig. Ausnahme: Man will die Least-Cost-Router-Features von fli4l nutzen. Dann sind sämtliche erlaubten Circuits für verschiedene Zeitbereiche zu definieren, siehe unten.

ISDN_CIRCUITS_N Gibt die Anzahl der verwendeten ISDN-Circuits an. Wird fli4l lediglich als ISDN-Anrufmonitor eingesetzt, ist einzustellen:

```
ISDN_CIRCUITS_N='0'
```

Soll der fli4l-Router lediglich als einfaches ISDN-Gateway in das Internet verwendet werden, reicht ein Circuit. Ausnahme: LC-Routing, siehe unten.

ISDN_CIRC_x_NAME Hier sollte ein Name für den Circuit vergeben werden - max. 15 Stellen lang. Dieser wird dann im imon-Client `imonc.exe` statt der angewählten Telefonnummer gezeigt. Erlaubte Zeichen sind die Buchstaben 'A' bis 'Z' (Klein- und Großschreibung), die Zahlen '0' bis '9' und der Bindestrich '-'., wie z.B.

```
ISDN_CIRC_x_NAME='msn'
```

Der Name kann außerdem im Paketfilter oder bei OpenVPN benutzt werden. Wenn z.B. der Paketfilter einen ISDN Circuit regeln soll, muß ein 'circuit_' dem Circuit Namen vorangesetzt werden. Heißt ein ISDN Circuit z.B. 'willi', so wird daraus folgendes im Paketfilter:

```
PF_INPUT_3='if:circuit_willi:any prot:udp 192.168.200.226 192.168.200.254:53 ACCEPT'
```

ISDN_CIRC_x_USEPEERDNS Hiermit wird festgelegt, ob die vom Internet-Provider bei der Einwahl übergebenen Nameserver für die Dauer der Onlineverbindung in die Konfigurationsdatei des lokalen Nameservers eingetragen werden sollen. Sinnvoll ist die Nutzung dieser Option also nur bei Circuits für Internet-Provider. Inzwischen unterstützen fast alle Provider diese Art der Übergabe.

Nachdem die Nameserver-IP-Adressen übertragen wurden, werden die in der base.txt unter *DNS_FORWARDERS* eingetragenen Nameserver aus der Konfigurationsdatei des lokalen Nameservers entfernt und die vom Provider vergebenen IP-Adressen als Forwarder eingetragen. Danach wird der lokale Nameserver veranlaßt, seine Konfiguration neu einzulesen. Dabei gehen bis dahin aufgelöste Namen nicht aus dem Nameserver-Cache verloren.

Diese Option bietet den Vorteil, immer mit den am nächsten liegenden Nameservern arbeiten zu können, sofern der Provider die korrekten IP-Adressen übermittelt - dadurch geht die Namensauflösung schneller.

Im Falle eines Ausfalls eines DNS-Servers beim Provider werden in der Regel die übergebenen DNS-Server-Adressen sehr schnell vom Provider korrigiert.

Trotz allem ist vor jeder ersten Einwahl die Angabe eines gültigen Nameservers in *DNS_FORWARDERS* der base.txt zwingend erforderlich, da sonst die erste Anfrage nicht korrekt aufgelöst werden kann. Außerdem wird beim Beenden der Verbindung die originale Konfiguration des lokalen Nameservers wieder hergestellt.

Standard-Einstellung: `ISDN_CIRC_x_USEPEERDNS`='yes'

ISDN_CIRC_x_TYPE `ISDN_CIRC_x_TYPE` gibt den Typ der x-ten IP-Verbindung an. Dabei sind folgende Werte möglich:

'raw' RAW-IP
'ppp' Sync-PPP

In den meisten Fällen wird PPP verwendet, jedoch ist Raw-IP etwas effizienter, da hier der PPP-Overhead entfällt. Eine Authentifizierung ist zwar bei Raw-IP nicht möglich, es kann jedoch über die Variable `ISDN_CIRC_x_DIALIN` (s.u.) eine Zugangsbeschränkung auf ganz bestimmte ISDN-Nummern (Stichwort "Clip") eingestellt werden. Wird `ISDN_CIRC_x_TYPE` auf 'raw' gestellt wird analog zu den PPP up/down Scripten in /etc/ppp ein raw up/down Script ausgeführt.

ISDN_CIRC_x_BUNDLING Für die ISDN-Kanalbündelung wird das verbreitete MPPP-Protokoll nach RFC 1717 verwendet. Damit gelten folgende Einschränkungen, die aber in der Praxis meist nicht relevant sind:

- Nur mit PPP-Verbindungen möglich, nicht bei Raw-Circuits
- Kanalbündelung nach neuerer RFC 1990 (MLP) ist nicht möglich

Der 2. Kanal kann entweder mit dem Client imonc manuell hinzugeschaltet werden oder über die Bandbreitenanpassung automatisch aktiviert werden, siehe auch die Beschreibung zu `ISDN_CIRC_x_BANDWIDTH`.

Standard-Einstellung: `ISDN_CIRC_x_BUNDLING`='no'

Vorsicht: bei Verwendung von Kanalbündelung zusammen mit Kompression kann es zu Problemen kommen, siehe auch die Beschreibung zu `ISDN_CIRC_x_FRAMECOMP`.

ISDN_CIRC_x_BANDWIDTH Ist die ISDN-Kanalbündelung über `ISDN_CIRC_x_BUNDLING`='yes' aktiviert, kann mit dieser Variablen eine automatische Hinzuschaltung des 2. ISDN-Kanals eingestellt werden. Dabei sind 2 numerische Parameter anzugeben:

1. Schwellenwert in Byte/Sekunde (S)
2. Zeitintervall in Sekunden (Z)

Wird der Schwellenwert S für Z Sekunden überschritten, schaltet der Steuerprozeß imond den 2. Kanal automatisch hinzu. Wird der Schwellenwert S für Z Sekunden unterschritten, wird der 2. Kanal automatisch wieder deaktiviert. Die automatische Bandbreitenanpassung kann mit `ISDN_CIRC_1_BANDWIDTH`=" abgeschaltet werden. Dann ist lediglich eine manuelle Kanalbündelung über den Client imonc möglich.

Beispiele:

- `ISDN_CIRC_1_BANDWIDTH='6144 30'`

  Überschreitet die Transferrate den Wert von 6 kibibyte/second für 30 Sekunden, wird der 2. Kanal hinzugeschaltet.

- `ISDN_CIRC_1_BANDWIDTH='0 0'`

  Der zweite ISDN-Kanal wird sofort, spätestens 10 Sekunden nach dem Verbindungsaufbau hinzugeschaltet und bleibt solange bestehen, bis die komplette Verbindung beendet wird.

- `ISDN_CIRC_1_BANDWIDTH=''`

  Der zweite ISDN-Kanal kann nur manuell hinzugeschaltet werden, Voraussetzung ist jedoch weiterhin, daß `ISDN_CIRC_1_BUNDLING='yes'` eingestellt ist.

- `ISDN_CIRC_1_BANDWIDTH='10000 30'`

  Eigentlich sollte hiermit der 2. Kanal nach 30 Sekunden hinzugeschaltet werden, wenn während dieser Zeitspanne 10000 B/s erreicht werden. Dazu wird es aber nicht kommen, da mit ISDN nicht mehr als 8 kB/s erreicht werden können.

Ist `ISDN_CIRC_x_BUNDLING='no'` eingestellt, ist der Wert in der Variablen `ISDN_CIRC_x_BANDWIDTH` belanglos.

Standard-Einstellung: `ISDN_CIRC_x_BANDWIDTH=''`

ISDN_CIRC_x_LOCAL In dieser Variablen wird die lokale IP-Adresse auf der ISDN-Seite hinterlegt.

Bei dynamischer Adresszuweisung sollte dieser Wert **leer** sein. Beim Verbindungsaufbau wird dann die IP-Adresse ausgehandelt. In den meisten Fällen vergeben Internet-Provider diese Adresse dynamisch. Soll jedoch eine fest vergebene IP-Adresse verwendet werden, ist diese hier einzutragen. Diese Variable ist optional und muss bei Bedarf in das Konfigfile eingetragen werden.

ISDN_CIRC_x_REMOTE In dieser Variablen wird die entfernte IP-Adresse und die zugehörige Netzmaske auf der ISDN-Seite hinterlegt. Dazu muss die CIDR (Classles Inter-Domain Routing) Schreibweise benutzt werden. Details zu CIDR (Seite 39) ist in der Dokumentation des Baseispaketes bei IP_NET_x zu finden.

Bei dynamischer Adresszuweisung sollte dieser Wert **leer** sein. Beim Verbindungsaufbau wird dann die IP-Adresse ausgehandelt. In den meisten Fällen vergeben Internet-Provider diese Adresse dynamisch. Soll jedoch eine fest vergebene IP-Adresse verwendet werden, ist diese hier einzutragen. Diese Variable ist optional und muss bei Bedarf in das Konfigfile eingetragen werden.

Die angegebene Netzmaske wird bei der Configuration des Interfaces nach der Einwahl verwendet. Während dieser Configuration wird auch eine Route zum Host erzeugt, in den man sich einwählt. Da man diese Route in der Regel nicht braucht, ist es günstig, hier nur eine Route direkt zum Einwahlrechner zu erzeugen. Dazu setzt man die Netzmaske auf /32, indem man hier 32 als Anzahl der gesetzten Bits in der Netzmaske spezifiziert. Für Details siehe Kapitel: Technische Details zum Dialin (Seite 372).

ISDN_CIRC_x_MTU ISDN_CIRC_x_MRU Mit diesen optionalen Variablen können die sog. **MTU** (maximum transmission unit) und die **MRU** (maximum receive unit) eingestellt werden. Optional bedeutet, die Variable muß nicht in der Konfigurationsdatei

stehen, sie ist bei Bedarf durch den Benutzer einzufügen!
Normal beträgt die MTU 1500 und die MRU 1524. Diese Einstellung sollte nur in Sonderfällen geändert werden!

ISDN_CIRC_x_CLAMP_MSS Hier sollte man ein yes setzen, wenn man synchrones ppp verwendet (`ISDN_CIRC_x_TYPE`='ppp') und eines der folgenden Symptome auftritt:

- der Webbrowser verbindet sich mit dem Webserver, aber es wird keine Seite angezeigt und kommt auch keine Fehlermeldung; es passiert einfach nichts mehr,
- das Versenden kleiner E-Mails funktioniert, bei großen gibt es Probleme oder
- ssh funktioniert, scp hängt nach dem initialen Verbindungsaufbau.

Provider, bei denen solche Probleme auftreten, sind z.B. Compuserve und andere Mediaways basierte Zugänge.

Standard-Einstellung: `ISDN_CIRC_x_CLAMP_MSS`='no'

ISDN_CIRC_x_HEADERCOMP Mit `ISDN_CIRC_x_HEADERCOMP`='yes' kann die Van-Jacobson-Komprimierung oder Headerkomprimierung eingestellt werden. Nicht alle Provider unterstützen das. Sollte es daher bei eingeschalteter Komprimierung zu Einwahlproblemen kommen, sollte `ISDN_CIRC_x_HEADERCOMP`='no' eingestellt werden.

Standard-Einstellung: `ISDN_CIRC_x_HEADERCOMP`='yes'

ISDN_CIRC_x_FRAMECOMP (EXPERIMENTAL) Dieser Parameter wird nur berücksichtigt, wenn `OPT_ISDN_COMP`='yes' eingestellt wird. Er regelt die Frame-Komprimierung.

Folgende Werte sind möglich:

'no'	Keine Frame-Komprimierung
'default'	LZS according RFC1974(std) and BSDCOMP 12
'all'	Negotiate lzs and bsdcomp
'lzs'	Negotiate lzs only
'lzsstd'	LZS according RFC1974 Standard Mode ("Sequential Mode")
'lzsext'	LZS according RFC1974 Extended Mode
'bsdcomp'	Negotiate bsdcomp only
'lzsstd-mh'	LZS Multihistory according RFC1974 Standard Mode ("Sequential Mode")

Welcher Wert für den jeweiligen Provider verwendet werden kann, muss ausprobiert werden. So weit bekannt geht bei T-Online nur 'lzsext'. Bei den meisten anderen Providern sollte man mit 'default' auskommen.

Vorsicht: Bei verwendung von Kanalbündelung in zusammenhang mit 'lzsext' kann es zu Problemen kommen. Diese Probleme sind, so weit bekannt, Einwahlserverspezifisch und damit meistens Providerspezifisch. Es können aber bei einem Provider auch verschiedene Typen Einwahlserver im Einsatz sein, es kann in dem Fall zu Unterschieden zwischen Einwahlknoten kommen.

'lzsstd-mh' ist für Router-zu-Routerbetrieb (r2r) gedacht. Das Verfahren wird von Providern nicht eingesetzt aber bringt bei Verwendung zwischen zwei FLI4L-Router erhebliche Verbesserung bei gleichzeitigen Übertragung von mehreren Dateien. Die Headerkompression ist dazu erforderlich und wird deshalb automatisch eingeschaltet.

ISDN_CIRC_x_REMOTENAME Diese Variable ist normalerweise lediglich bei der Konfiguration von fli4l als Einwahlrouter relevant. Hier kann ein Name des Remote-Hosts eingetragen werden, muß aber nicht.

Standard-Einstellung: `ISDN_CIRC_x_REMOTENAME=”`

ISDN_CIRC_x_PASS Hier sind die Provider-Daten einzutragen. Im Beispiel oben handelt es sich um die Daten des Providers Microsoft Network.

`ISDN_CIRC_x_USER` enthält die Benutzerkennung, `ISDN_CIRC_x_PASS` das Password.

WICHTIG: Für einen T-Online-Zugang ist folgendes zu beachten:

Der Username AAAAAAAAAAAATTTTTT#MMMM setzt sich aus der zwölfstelligen Anschlußkennung, der T-Online-Nummer und der Mitbenutzernummer zusammen. Hinter der T-Online-Nummer muß ein ’#’ angegeben werden, wenn die Länge der T-Online-Nummer kürzer als 12 Zeichen ist.

Sollte dies in Einzelfällen nicht zum Erfolg führen (offenbar abhängig von der Vermittlungsstelle), muß zusätzlich zwischen der Anschlußkennung und der T-Online-Nummer ein weiteres ’#’-Zeichen eingefügt werden.

Ansonsten (T-Online-Nr ist 12stellig) sind keine ’#’-Zeichen anzugeben.

Beispiel: `ISDN_CIRC_1_USER=’123456#123’`

Bei Raw-IP-Circuits haben diese Variablen keine Bedeutung.

ISDN_CIRC_x_ROUTE_N Die Anzahl der Routen die dieser ISDN Circuit bedient. Wenn dieser Circuit eine Default-Route definiert muss der Eintrag auf ’1’ gesetzt werden.

ISDN_CIRC_x_ROUTE_X Die Route oder die Routen für diesen Circuit. Für einen Internet-Zugang sollte man hier im ersten Eintrag ’0.0.0.0/0’ (default route) angeben. Das Format ist immer ’network/netmaskbits’. Eine Hostroute würde z.B. so aussehen: ’192.168.199.1/32’. Bei Einwahl in den Firmen- oder Uni-Router ist lediglich das oder die Netze anzugeben, die man dort erreichen will, z.B.

```
ISDN_CIRC_%_ROUTE_N=’2’
ISDN_CIRC_%_ROUTE_1=’192.168.8.0/24’
ISDN_CIRC_%_ROUTE_2=’192.168.9.0/24’
```

Bei mehreren Netzen müssen diese jeweils in einen eigenen Eintrag, also für jede Route muss eine ISDN_CIRC_x_ROUTE_y=” Zeile angelegt werden.

Möchte man die LC-Routing-Features von fli4l nutzen, kann *mehreren* Circuits eine Default-Route zugewiesen werden. Welcher Circuit dann tatsächlich verwendet wird, wird über `ISDN_CIRC_x_TIMES` eingestellt, siehe unten.

ISDN_CIRC_x_DIALOUT `ISDN_CIRC_x_DIALOUT` gibt die zu wählende Telefonnummer an. Es ist möglich, hier mehrere Nummern anzugeben (falls eine besetzt ist, wird die nächste angewählt) - die Trennung erfolgt dabei durch Leerzeichen. Laut Berichten in der Newsgroup dürfen maximal fünf Nummern angegeben werden.

ISDN_CIRC_x_DIALIN Soll der Circuit (auch) zum Einwählen genutzt werden, wird in ISDN_CIRC_x_DIALIN die Rufnummer des Anrufenden eingesetzt - und zwar mit Vorwahl, aber *ohne* die erste 0. Bei Anschlüssen hinter Telefonanlagen kann dies anders sein, eventuell müssen dann eine oder sogar zwei führende Nullen angegeben werden.

Soll es mehreren Teilnehmern ermöglicht werden, sich über diesen Circuit einzuwählen, können diese Nummern durch Leerzeichen getrennt angegeben werden. Besser ist es aber, jedem Gegner einen extra Circuit zuzuweisen. Sonst könnte es bei Einwahl von zwei Gegnern zu gleicher Zeit (ist über die 2 ISDN-Kanäle durchaus möglich) auf demselben Circuit zu Kollisionen bzgl. IP-Adressen kommen.

Falls der Anrufende keine Rufnummer überträgt, hier eine '0' setzen. Aber Vorsicht: damit wird jedem eine Anwahl gestattet, der keine Rufnummer überträgt!

Möchte man eine Einwahl unabhängig von der MSN des Anrufenden realisieren, ist als Wert '*' anzugeben.

In den beiden letzten Fällen ist ein Authentifizierungsverfahren (siehe ISDN_CIRC_x_AUTH) unumgänglich.

ISDN_CIRC_x_CALLBACK Einstellung Callbackverfahren, mögliche Werte:

'in'	fli4l wird angerufen und ruft zurück
'out'	fli4l ruft an, hängt jedoch wieder ein und wartet auf Rückruf
'off'	kein Callback
'cbcp'	CallBack Control Protocol
'cbcp0'	CallBack Control Protocol 0
'cbcp3'	CallBack Control Protocol 3
'cbcp6'	CallBack Control Protocol 6

Bei den CallBack Control Protokolle (auch 'Microsoft CallBack' genannt) ist cbcp6 das meist übliche Protokoll.

Standard-Wert: 'off'

ISDN_CIRC_x_CBNUMBER Hier kann man für die Protokolle cbcp, cbcp3 und cbcp6 eine Rückrufnummer einsetzen (bei cbcp3 Pflicht).

ISDN_CIRC_x_CBDELAY Diese Variable gibt eine Verzögerung in Sekunden an, die bei Callback gewartet werden soll. Je nachdem, in welcher Richtung der Callback erfolgen soll, hat diese Variable eine etwas andere Bedeutung:

- ISDN_CIRC_x_CALLBACK='in':

  Wird fli4l angerufen und soll zurückrufen, ist ISDN_CIRC_x_CBDELAY die Wartezeit, bis der Rückruf erfolgen soll. Ein guter Erfahrungswert ist ISDN_CIRC_x_CBDELAY='3'. Je nach "Gegner" kann aber auch ein geringerer Wert funktionieren, welches dann den Verbindungsaufbau beschleunigen kann.

- ISDN_CIRC_x_CALLBACK='out':

  In diesem Fall gibt ISDN_CIRC_x_CBDELAY die Zeit an, wie lange das "Anklingeln des Gegners" erfolgen soll, bis auf den Rückruf gewartet wird. Auch hier ist ISDN_CIRC_x_CBDELAY='3' ein guter Erfahrungswert. Was mir dazu aufgefallen ist: Bei Fernverbindungen muß man bis zu 3 Sekunden "klingeln" lassen, bevor der andere

Router überhaupt den Anruf sieht. Bei Ortsverbindungen kann meist dieser Wert kleiner gewählt werden. Im Zweifel: Ausprobieren!

Ist die Variable ISDN_CIRC_x_CALLBACK='off' eingestellt, wird ISDN_CIRC_x_CBDELAY ignoriert. Auch beim CallBack Control Protocol hat diese Variable keine Bedeutung.

ISDN_CIRC_x_EAZ Im Beispiel ist die MSN (hier EAZ genannt) auf 81330 gesetzt. Hier sollte die eigene MSN *ohne* Vorwahl eingetragen werden.

Bei Anschlüssen hinter einer Telefonanlage mit Anlagenanschluss ist meistens nur die Durchwahl anzugeben. Ich habe aber auch schon gelesen, dass eine '0' weiterhelfen kann, wenn es Probleme mit der verwendeten Telefonanlage geben sollte.

Dieses Feld kann auch leer sein. Dies soll bei besonders hartnäckigen TK-Anlagen helfen. Um Feedback wird an dieser Stelle gebeten.

ISDN_CIRC_x_SLAVE_EAZ Ist der fli4l-Router am internen S0-Bus einer Telefonanlage angeschlossen und möchte man Kanalbündelung verwenden, ist bei manchen Telefonanlagen die Angabe einer 2. Durchwahlnummer für den Slave-Kanal einzutragen.

Im Normalfall kann diese Variable jedoch leer bleiben.

ISDN_CIRC_x_DEBUG Soll ipppd zusätzliche Debug-Informationen ausgeben, muss man ISDN_CIRC_x_DEBUG auf 'yes' setzen. In diesem Fall schreibt ipppd zusätzlichen Informationen über die syslog-Schnittstelle.

WICHTIG: Damit diese auch über syslogd ausgegeben werden, muss die Variable OPT_SYSLOGD (Siehe OPT_SYSLOGD - Programm zum Protokollieren von Systemfehlermeldungen (Seite 81)) ebenso auf 'yes' gesetzt sein.
Weil manche Meldungen über klog ausgegeben werden sollte man beim Debuggen von ISDN auch OPT_KLOGD (Siehe OPT_KLOGD - Kernel-Message-Logger (Seite 83)) auf 'yes' setzen.

Bei Raw-IP-Circuits hat ISDN_CIRC_x_DEBUG keine Bedeutung.

ISDN_CIRC_x_AUTH Wird dieser Circuit auch zum Einwählen verwendet und soll eine Authentifizierung über PAP oder CHAP vom einwählenden "Gegner" gefordert werden, ist ISDN_CIRC_x_AUTH auf 'pap' oder 'chap' zu setzen - und *nur* dann. Anderenfalls immer leer lassen!

Grund: Ein angewählter Internet-Provider wird es immer ablehnen, sich selbst auszuweisen! Ausnahmen bestätigen die Regel, wie ich erst kürzlich in der i4l-Mailingliste las ...

Als Benutzername und Passwort werden die Einträge von ISDN_CIRC_x_USER und ISDN_CIRC_x_PASS benutzt.

Bei Raw-IP-Circuits hat diese Variable keine Bedeutung.

ISDN_CIRC_x_HUP_TIMEOUT Mit der Variablen ISDN_CIRC_x_HUP_TIMEOUT wird die Zeit gesteuert, nach der der fli4l-Rechner die Verbindung zum Provider beenden soll, wenn nichts mehr über die Leitung geht. Im Beispiel wird die Verbindung nach 40 Sekunden Idle-Time abgebaut, um Geld zu sparen. Bei einem erneuten Zugriff in's Netz wird die

Verbindung in Sekundenschnelle wieder aufgebaut. Sinnvoll bei Providern, die sekundengenau abrechnen!

Man sollte zumindest in der Testphase das automatische Wählen/Einhängen des fli4l-Routers beobachten (entweder auf der Console oder im imon-Client für Windows). Nicht, dass durch eine fehlerhafte Konfiguration der ISDN-Anschluss zur Standleitung wird.

Wird der Wert auf '0' gestellt, wird keine Idle-Zeit mehr berücksichtigt, d.h. fli4l hängt von sich aus die Leitung nicht mehr ein. Bitte mit Vorsicht anwenden.

Auch habe ich die Erfahrung gemacht, dass beim Beenden von Netscape auf einem Rechner im lokalen Netz noch einmal rausgewählt wird, wenn zwischendurch bereits eingehängt wurde. Offenbar hat Netscape da noch eine Verbindung zum zuletzt gewählten Web-Server offen, die er erst beim Beenden schließt. Und dieses Schließen führt zum Wählen ...

ISDN_CIRC_x_CHARGEINT Charge-Interval: Hier ist der Zeittakt in Sekunden anzugeben. Dieser wird dann für die Kosten-Berechnung verwendet.

Die meisten Provider rechnen minutengenau ab. In diesem Fall ist der Wert '60' richtig. Compuserve verwendet einen 3-Minuten-Takt (Stand Juni 2000), also `ISDN_CIRC_x_CHARGEINT`='180'. Bei Providern mit sekundengenauer Abrechnung (z.B. Planet-Interkom) setzt man besser `ISDN_CIRC_x_CHARGEINT` auf '1'.

Erweiterung für `ISDN_CIRC_x_CHARGEINT` >= 60 Sekunden:

Wurde `ISDN_CIRC_x_HUP_TIMEOUT` Sekunden lang kein Traffic bemerkt, wird ca. 2 Sekunden vor Ablauf des Taktes eingehängt. Die vom Provider berechnete Zeit wird also fast komplett ausgenutzt. Ein wirklich tolles Feature von isdn4linux!

Bei sekundengenau abgerechneten Verbindungen hat das natürlich keinen Sinn - daher gilt diese Regelung erst ab Zeittakten von 60 Sekunden.

ISDN_CIRC_x_TIMES Hier werden die Zeiten angegeben, wann dieser Circuit aktiviert werden soll und wann er wieviel kostet. Dadurch wird es möglich, zu verschiedenen Zeiten verschiedene Circuits mit Default-Routen zu verwenden (Least-Cost-Routing). Dabei kontrolliert der Daemon imond die Routen-Zuweisung.

Aufbau der Variablen:

```
ISDN_CIRC_x_TIMES='times-1-info [times-2-info] ...'
```

Jedes Feld times-?-info besteht aus 4 Unterfeldern - durch Doppelpunkt (':') getrennt.

1. Feld: W1-W2

   Wochentag-Zeitraum, z.B. Mo-Fr oder Sa-Su usw. Schreibweise in Englisch oder deutsch möglich. Soll ein einzelner Wochentag eingetragen werden, ist zu W1-W1 schreiben, also z.B. Su-Su.

2. Feld: hh-hh

   Stunden-Bereich, z.B. 09-18 oder auch 18-09. 18-09 ist gleichbedeutend mit 18-24 plus 00-09. 00-24 meint den ganzen Tag.

3. Feld: Charge

   Hier werden in Euro-Werten die Kosten pro Minute angegeben, z.B. 0.032 für 3.2 Cent pro Minute. Diese werden unter Berücksichtigung der Taktzeit umgerechnet für die tatsächlich anfallenden Kosten, welche dann im imon-Client angezeigt werden.

4. Feld: LC-Default-Route

   Der Inhalt kann Y oder N sein. Dabei bedeutet:

   - Y: Der angegebene Zeitbereich wird beim LC-Routing als Default-Route verwendet. Wichtig: In diesem Fall muss auch `ISDN_CIRC_x_ROUTE`='0.0.0.0/0' sein!
   - N: Der angegebene Zeitbereich dient nur zum Berechnen von Kosten, er wird beim automatischen LC-Routing jedoch nicht weiter verwendet.

Beispiel:

```
ISDN_CIRC_1_TIMES='Mo-Fr:09-18:0.049:N Mo-Fr:18-09:0.044:Y Sa-Su:00-24:0.044:Y'
ISDN_CIRC_2_TIMES='Mo-Fr:09-18:0.019:Y Mo-Fr:18-09:0.044:N Sa-Su:00-24:0.044:N'
```

Die Bedeutung ist dabei wie folgt: Circuit 1 (Provider Planet-Interkom) soll abends an Werktagen und komplett am Wochenende verwendet werden, jedoch tagsüber an Werktagen der Circuit 2 (Provider Compuserve).

Wichtig: *Die bei `ISDN_CIRC_x_TIMES` angegebenen Zeiten muessen die ganze Woche abdecken. Ist das nicht der Fall, kann keine gültige Konfiguration erzeugt werden.*

Wenn die Zeitbereiche aller LC-Default-Route-Circuits ("Y") zusammengenommen nicht die komplette Woche beinhalten, gibt's zu diesen Lückenzeiten keine Default-Route. Damit ist dann Surfen im Internet zu diesen Zeiten ausgeschlossen!

Beispiel:

```
ISDN_CIRC_1_TIMES='Sa-Su:00-24:0.044:Y Mo-Fr:09-18:0.049:N Mo-Fr:18-09:0.044:N'
ISDN_CIRC_2_TIMES='Sa-Su:00-24:0.044:N Mo-Fr:09-18:0.019:Y Mo-Fr:18-09:0.044:N'
```

Hier wurde für die Werktage von 18-09 Uhr "N" gesetzt. Zu diesen Zeiten gibt es keine Route in's Internet: Surfen verboten!

Noch ein ganz einfaches Beispiel:

```
ISDN_CIRC_1_TIMES='Mo-Su:00-24:0.0:Y'
```

für diejenigen, die eine Flatrate nutzen.

Und noch eine letzte Bemerkung zum LC-Routing:

Deutsche Feiertage werden wie Sonntage behandelt.

4.12.5. OPT_TELMOND - telmond-Konfiguration

Mit `OPT_TELMOND` kann man einstellen, ob der telmond-Server aktiviert werden soll. Dieser horcht auf eingehende Telefonanrufe und teilt über TCP-Port 5001 die anrufende und die angerufene Telefonnummer mit. Diese Information kann vom Windows- und Unix/Linux-Client imonc abgefragt und angezeigt werden (s.a. Kapitel "Client-/Server-Schnittstelle imond").

Zwingende Voraussetzung ist hierfür die Installation einer ISDN-Karte und die Konfiguration von OPT_ISDN und den dazugehörenden Konfigurations-Variablen.

Im laufenden Betrieb kann die korrekte Funktion von telmond unter Linux/Unix/Windows überprüft werden mit:

```
telnet fli4l 5001
```

Dann sollte der letzte Anruf gezeigt und anschließend die telnet-Verbindung sofort wieder geschlossen werden.

Der Port 5001 ist nur vom LAN aus erreichbar. Standardmäßig wird ein Zugriff von außen über die Firewall-Konfiguration abgeblockt. Möchte man jedoch auch den Zugriff innerhalb des LANs anders regeln, kann dies über die weitere telmond-Konfigurationsvariablen eingestellt werden, siehe unten.

Standard-Einstellung: START_TELMOND='yes'

TELMOND_PORT TCP/IP-Port, auf dem telmond auf Verbindungen horcht. Der Standardwert '5001' sollte nur in Ausnahmefällen geändert werden.

TELMOND_LOG Bei TELMOND_LOG='yes' werden sämtliche einkommenden Telefonanrufe in der Datei /var/log/telmond.log gespeichert. Der Inhalt der Datei kann mit dem imond-Client imonc unter Unix/Linux und Windows abgefragt werden.

Abweichende Pfade bzw. nach Clients aufgeteilte Log-Dateien können weiter unten konfiguriert werden.

Standard-Einstellung: TELMOND_LOG='no'

TELMOND_LOGDIR Ist das Protokollieren eingeschaltet, kann über TELMOND_LOGDIR ein alternatives Verzeichnis statt /var/log angegeben werden, z.B. '/boot'. Dann wird die LOG-Datei telmond.log auf der Diskette angelegt. Dazu muß dann die Diskette auch Read/Write "gemounted" sein.

TELMOND_MSN_N Sollen bestimmte Anrufe nur auf einigen PC-Clients im imonc sichtbar werden, kann ein Filter eingestellt werden, mit dem Anrufe auf spezielle MSNs nur für diese PC-Clients protokolliert werden.

Ist so etwas notwendig, z.B. in einer WG, wird die Variable TELMOND_MSN_N auf die Anzahl der MSN-Filter eingestellt.

Standard-Einstellung: TELMOND_MSN_N='0'

TELMOND_MSN_x Für jeden MSN-Filter ist eine Liste von IP-Adressen anzugeben, für welche die Anrufe auf eingetragene MSN sichtbar werden sollen.

Die Variable TELMOND_MSN_N bestimmt die Anzahl solcher Konfigurationen, siehe oben.

Der Aufbau der Variblen ist:

```
TELMOND_MSN_x='MSN IP-ADDR-1 IP-ADDR-2 ...'
```

Ein einfaches Beispiel:

```
TELMOND_MSN_1='123456789 192.168.6.2'
```

Soll ein Anruf auf eine bestimmte MSN auf mehreren Rechnern sichtbar werden, z.B. Fax, sind die IP-Adressen der Rechner hintereinander anzugeben, z.B.

```
TELMOND_MSN_1='123456789 192.168.6.2 192.168.6.3'
```

TELMOND_CMD_N Sobald ein Telefonanruf (Voice) auf einer bestimmten MSN hereinkommt, können optional bestimmte Kommandos auf dem fli4l-Router ausgeführt werden. Mit `TELMOND_CMD_N` gibt man die Anzahl der konfigurierten Kommandos an.

TELMOND_CMD_x Mit `TELMOND_CMD_1` bis `TELMOND_CMD_n` können Kommandos angegeben werden, welche ausgeführt werden, wenn ein Telefonanruf eintrifft.

Die Variable `TELMOND_CMD_N` bestimmt die Anzahl solcher Kommandos, siehe oben.

Die Variable hat folgenden Aufbau:

```
MSN CALLER-NUMBER  COMMAND ...
```

Dabei ist die MSN ohne Vorwahl einzutragen. Als CALLER-NUMBER gibt man die komplette Telefonnummer - also mit Vorwahl - an. Schreibt man als Wert für CALLER-NUMBER ein einfaches Sternchen (*), wird von telmond die Telefonnummer des Anrufers nicht ausgewertet.

Hier ein Beispiel:

```
TELMOND_CMD_1='1234567 0987654321 sleep 5; imonc dial'
TELMOND_CMD_2='1234568 * switch-on-coffee-machine'
```

Im ersten Fall wird die Kommandofolge "sleep 5; imonc dial" durchgeführt, wenn der Anrufer mit der Telefonnummer 0987654321 die MSN 1234567 anruft. Tatsächlich werden hier 2 Kommandos ausgeführt. Zunächst wird 5 Sekunden gewartet, damit der ISDN-Kanal wieder frei wird, auf dem der Anruf hereinkam. Anschließend wird der fli4l-Client imonc mit dem Argument "dial" gestartet. imonc gibt dieses Kommando 1:1 an den Server imond weiter, welcher dann auf dem Default-Circuit eine Netzverbindung herstellt, z.B. ins Internet. Welche Kommandos das imonc-Client-Programm an den Server imond weitergeben kann, ist im Kapitel "Client-/Server-Schnittstelle imond" erklärt. Damit diese Einstellung funktioniert, muss `OPT_IMONC` aus dem Paket "tools" installiert sein.

Das zweite Kommando "switch-on-coffee-machine" wird ausgeführt, wenn ein Anruf auf der MSN 1234568 hereinkommt, unabhängig, woher der Anruf kam. Natürlich gibt es das Kommando "switch-on-coffee-machine" (noch) nicht für fli4l!

Beim Aufruf der Kommandos können folgende Platzhalter verwendet werden:

%d	date	Datum
%t	time	Uhrzeit
%p	phone	Telefonnummer des Anrufers
%m	msn	Eigene MSN
%%	percent	das Prozentzeichen selbst

Diese Daten können dann von den aufgerufenen Programmen weiter verwendet werden, z.B. zum Verschicken per E-Mail.

4.13. LCD - Anzeige von Statusinformationen über LC-Display

4.13.1. Einleitung

Mit diesem Paket ist es möglich, ein LCD-Modul an den Parallelport des fli4l Rechners anzuschließen. Auch können nun serielle LCD-Module der Firma Matrix-Orbital verwendet werden. Es gibt ausserdem Filter für spezielle Displays.

Auf diesem Display werden Informationen wie Datum, Uhrzeit, die aktuellen Loadwerte und natürlich auch der ISDN- oder DSL-Durchsatz für Up- und Download in kb/s und ein Balken angegeben.

4.13.2. Konfiguration

Möchte man das LCD-Paket einsetzen, sind noch folgende Variablen zu setzen:

```
OPT_LCD='yes'  (Standard-Einstellung: OPT_LCD='no')
```

LCD_COLS - Anzahl der Zeichen pro Zeile. Das Kernelmodul unterstützt momentan Werte von 16, 20, 24, 32 und 40. Module mit 8 oder 27 Zeichen funktionieren üblicherweise ebenfalls. Am Kernelmodul wird dann von fli4l 16 bzw. 40 übergeben.

LCD_LINES - Anzahl der Zeilen.

Mögliche Werte: 1, 2 und 4.

Achtung: Displays am Parallelport mit zwei Controllerchips (4x16, 4x40 etc.) müssen hier mit 2 Zeilen definiert werden! Die Koordinaten der anzuzeigenden Werte können normal angegeben werden. Der LCD-Treiber entscheidet anhand der Koordinate und der Anzahl der Zeilen, welcher der beiden Kontroller anzusprechen ist.

LCD_ADDRESS Die IO-Adresse des LPT-Ports, z.B. '0x278'

Wird ein serielles Display von Matrix-Orbital verwendet, ist hier die verwendete serielle Schnittstelle einzutragen, z.B. 'com1' oder 'com2'. Mit 'none' kann man LAN-Only einstellen, siehe: `LCD_LANIP`

Es ist möglich, die Ausgabe auf den Bildschirm auszugeben: 'console' oder tty1 wählen den Hauptbildschirm. Dies kann allerdings zur Vermischung mit den normalen Ausgaben führen und ist daher nicht zu empfehlen. 'tty2', 'tty3' ... 'tty9' selektieren andere virtuelle Konsolen, die mit ALT-F2 ... zu erreichen sind. ALT-F1 schaltet dann wieder auf den Hauptbildschirm zurück.

Wichtig: *Bisher wurden nur parallele Schnittstellen auf dem Mainboard oder auf ISA-Schnittstellenkarten unterstützt. PCI-Karten mit parallelen Schnittstellen konnten nicht verwendet werden. Diese Version hier erlaubt auch die Konfiguration von parallelen Schnittstellen auf bestimmten PCI-Karten mit NETMOS-Chips. Hier zu muss man sich mittels*

```
cat /proc/pci
```

die erkannten PCI-Geräte anzeigen lassen. Hier sucht man das Gerät mit der passenden Vendor-ID und Device-ID und wählt als io-Adresse den oder die folgenden Einträge aus:

- *Nm9705CV (Vendor id=9710, Device id=9705, Port1 1. Eintrag)*
- *Nm9735CV (Vendor id=9710, Device id=9735, Port1 3. Eintrag)*
- *Nm9805CV (Vendor id=9710, Device id=9805, Port1 1. Eintrag)*
- *Nm9715CV (Vendor id=9710, Device id=9815, Port1 1. Eintrag, Port2 3. Eintrag)*
- *Nm9835CV (Vendor id=9710, Device id=9835, Port1 3. Eintrag)*
- *Nm9755CV (Vendor id=9710, Device id=9855, Port1 1. Eintrag, Port2 3. Eintrag)*

Die Konfigurationsmöglichkeit wurde eingebaut, ohne entsprechende Hardware zum Testen zur Verfügung zu haben. Daher ist das als experimentelles Feature zu betrachten. Bei Fehlern bitte ausführliche Informationen in die Newsgroup posten!

LCD_LANIP (optionale Variable)

Die IP-Adresse eines 16x2 Displays an einem Pollin AVR-NET-IO im LAN oder andere Displays an ethersex - siehe weitere Variablen (experimentell)

LCD_LANTYPE (optionale Variable)

Typ der Firmware. Zur Auswahl stehen: 'pollin' (default) - Original-Firmware AVR-NET-IO 'ethersex' - Firmware von www.ethersex.de mit aktivem LCD

LCD_LANUSER (optionale Variable)

Authentifizierung am ethersex, wenn PAM auf ecmd/tcp konfiguriert wurde. Hier: Benutzername

LCD_LANPASS (optionale Variable)

Passwort zu `LCD_LANUSER`

LCD_TIME_LONG LCD_TIME_SHORT Dieses sind zwei Timer-Werte für den IO-Port zum LCD-Display. Werden die Variablen leer gelassen, gelten folgende Standardwerte:

```
LCD_TIME_LONG='100'
LCD_TIME_SHORT='40'
```

Sollte es Probleme mit dem LCD-Display geben, z.B. wenn wild verteilte Zeichen erscheinen, sollte man diese Werte erhöhen, z.B.

```
LCD_TIME_LONG='120'
LCD_TIME_SHORT='60'
```

Bei seriellen Displays von Matrix-Orbital sind diese Variablen ohne Bedeutung

LCD_ADDR_TYPE - Adressierungsart des LCD-Controllers.

```
LCD_ADDR_TYPE='0'     # Fuer HD44780 und kompatible Kontroller
LCD_ADDR_TYPE='1'     # Fuer HD66712 und kompatible Kontroller
LCD_ADDR_TYPE='2'     # Obsolet, Funktionalität ist jetzt
                      # in '0' eingebaut
```

Bei seriellen Displays von Matrix-Orbital ist diese Variable ohne Bedeutung

LCD_WINAMP Es existieren verschiedene Verdratungsvarianten bei fertig bestückten LCD-Displays von Kernel-Concepts, die normale und die Winamp-Verdrahtung. Neuere Displays kommen mit letzterer Variante, dann muss hier ein yes eingetragen werden.

LCD_FILTER - Filter für spezielle Displays. Zur Zeit gibt es Filter für - ipc_a78 Displays

```
LCD_FILTER='mo2ipc_a78'     # Fuer mo2ipc_a78 Displays
```

LCD_START_MSG Die hier eingetragene Meldung wird beim Systemstart kurz nach Laden der Treiber auf das Display ausgegeben. Sie sollte die Länge einer Zeile nicht überschreiten, da anderenfalls die vollständige Ausgabe des Textes nicht garantierte werden kann.

LCD_STOP_MSG Die hier eingetragene Meldung wird beim Herunterfahren des Systems auf das Display ausgegeben. Sie sollte die Länge einer Zeile nicht überschreiten, da anderenfalls die vollständige Ausgabe des Textes nicht garantierte werden kann.

LCD_REBOOT_MSG Die hier eingetragene Meldung wird beim Rebooten des Systems auf das Display ausgegeben. Sie sollte die Länge einer Zeile nicht überschreiten, da anderenfalls die vollständige Ausgabe des Textes nicht garantierte werden kann.

LCD_START_ISDN_RATE Gibt an, ob das Programm isdn_rate gestartet werden soll.

LCD_TYPE_N Ulf Lanz hat das Ausgabeformat im isdn_rate-Programm so variabel gestaltet, daß jeder Anwender die Anzeige seinen Wünschen gemäß zusammenstellen kann.

Bei `LCD_TYPE_N` ist die Anzahl der auszugebenden Datentypen anzugeben. Diese Datentypen werden immer angezeigt, egal ob man Online oder Offline ist.

LCD_TYPE_x `LCD_TYPE_x` gibt den Datentyp, die Spalte und die Zeile an, wo die gewünschte Information auf dem Display erscheinen soll. Der Datentyp ist numerisch codiert. Die möglichen Werte sind in Table 4.9 aufgeführt.

Die beiden nächsten Zahlen in `LCD_TYPE_1` geben die Position an. Format: "Spalte Zeile", wobei beide Zahlen bei 0 beginnen.

Beispiel:

```
LCD_TYPE_1='4 10 1'   # Status in der 2. Zeile, ab Spalte 11
           | |  |
           | |  \--    Zeile im Display
           | \-----    Spalte im Display
           \-------    Anzeigentyp laut Tabelle
```

Typ	Information	Zeichenbreite
0	local date dd.mm.yyyy	10
1	local date dd.mm.yy	8
2	local time hh:mm:ss	8
3	remote date dd.mm.yyyy	10
4	remote date dd.mm.yy	8
5	remote time hh:mm:ss	8
6	isdn status channel 1	7
7	isdn status channel 2	7
8	dsl status	7
9	isdn circuit name channel 1	16
10	isdn circuit name channel 2	16
11	dsl circuit name	16
12	isdn input rate bar	8
13	isdn output rate bar	8
14	dsl input rate bar	8
15	dsl output rate bar	8
16	isdn input rate	5
17	isdn output rate	5
18	dsl input rate	5
19	dsl output rate	5
20	isdn charge channel 1	6
21	isdn charge channel 2	6
22	dsl charge	6
23	isdn ip address channel 1	15
24	isdn ip address channel 2	15
25	dsl ip address	15
26	load 1	5
27	load 2	5
28	phone	16
29	isdn online time channel 1	8
30	isdn online time channel 2	8
31	dsl online time	8
32	isdn quantity in channel 1	8
33	isdn quantity in channel 2	8
34	dsl quantity in	8
35	isdn quantity out channel 1	8
36	isdn quantity out channel 2	8
37	dsl quantity out	8
38	cpu usage	4
39	fixed text	max 20
40	text -> /etc/lcd_text1.txt	max 20
41	text -> /etc/lcd_text2.txt	max 20
42	text -> /etc/lcd_text3.txt	max 20
43	text -> /etc/lcd_text4.txt	max 20
44	router uptime	10

Tabelle 4.9.: Übersicht über mögliche Werte für LCD_TYPE_x

Für Type 39 (fixed text) ist obiges Format noch um den Text zu erweitern der angezeigt werden soll.

Beispiel:

```
LCD_TYPE_2='39 10 1 Hallo'   # Text "Hallo" in der 2. Zeile
                             # ab Spalte 11
```

Die Typen 40 - 43 holen sich den Text, der angezeigt werden soll, aus den Dateien, die in der Typenliste erwähnt sind. Diese Dateien werden während der Laufzeit jede Sekunde eingelesen und angezeigt. Sie können von anderen Programmen (z.B. telmond) geändert werden. Damit kann man sich z.B. auf dem Display anzeigen lassen, dass man neue E-Mails hat, auch wenn man Offline ist (MyJack). Die Standardtexte für die Datentypen 40-43 werden mit folgenden Variablen definiert (und beim Starten des Systems in temporärren Dateien zwischengespeichert, deren Namen einfach immer durch Einfügen des Indizes in die Zeichenkette "/etc/lcd_text<Zahl>.txt" definiert wird):

```
LCD_VAR_TEXT_1='Text 1'  # -> /etc/lcd_text1.txt
LCD_VAR_TEXT_2='Text 2'  # -> /etc/lcd_text2.txt
LCD_VAR_TEXT_3='Text 3'  # -> /etc/lcd_text3.txt
LCD_VAR_TEXT_4='Text 4'  # -> /etc/lcd_text4.txt
```

Ab der Version 1.6.2 ist es auch möglich, in Abhängigkeit vom Onlinestatus verschiedene Texte anzeigen zu lassen. Z.B. kann nun während man Online ist, die aktuelle Onlinezeit angezeigt werden und wenn man Offline ist, wird an gleicher Stelle das Datum und die Uhrzeit angezeigt. Dafür sind folgende Variablen hinzugekommen:

LCD_TYPE_ONLINE_N Bei `LCD_TYPE_ONLINE_N` ist die Anzahl der auszugebenden Datentypen anzugeben. Diese Datentypen werden nur angezeigt wenn man Online ist.

LCD_TYPE_ONLINE_x `LCD_TYPE_ONLINE_x` gibt den Datentyp, die Spalte und die Zeile an, wo die gewünschte Information auf dem Display erscheinen soll. Der Datentyp ist numerisch codiert. Das Format und die Typen entsprechen denen aus der Tabelle bei 'LCD_TEXT_x'.

Beispiel:

```
LCD_TYPE_ONLINE_1=' 8  0 0'     # dsl status
```

LCD_TYPE_OFFLINE_N Bei `LCD_TYPE_OFFLINE_N` ist die Anzahl der auszugebenden Datentypen anzugeben. Diese Datentypen werden nur angezeigt wenn man Offline ist.

LCD_TYPE_OFFLINE_x `LCD_TYPE_OFFLINE_x` gibt den Datentyp, die Spalte und die Zeile an, wo die gewünschte Information auf dem Display erscheinen soll. Der Datentyp ist numerisch codiert. Das Format und die Typen entsprechen denen aus der Tabelle bei 'LCD_TEXT_x'.

Beispiel:

```
LCD_TYPE_OFFLINE_1=' 0  0 0'  # local date
```

LCD_DSL_SPEED_IN LCD_DSL_SPEED_OUT `LCD_DSL_SPEED_IN` und `LCD_DSL_SPEED_OUT` dienen zur Skalierung der Balkenanzeige (Typen 14 und 15). Hier werden die maximalen Übertragungsraten der DSL-Verbindung angegeben. Prinzipiell kan jeder ganzazahlige Wert angegeben werden. Es kann sogar sinnvoll sein, etwas höhere Werte anzugeben, um ein '+' in der letzten Stelle zu vermeiden. Bitte auch beachten, dass die tatsächliche Rate etwas höher als der Name des Anschlusses sugeriert, also DSL1000 hat eine Downloadrate (inbound) von 1024 kilobits/s.

Beispiel für DSL-Verbindung mit 1024/128 kilobit/s:

```
LCD_DSL_SPEED_IN='1024' # Bitrate for DSL inbound
LCD_DSL_SPEED_OUT='128' # Bitrate for DSL outbound
```

Für ISDN sind die Werte nicht relevant.

4.13.3. isdn_rate

Das Programm „isdn_rate“ ist der eigentliche Kern des LCD-Paketes. Es erfasst den Status der Circuits und gibt die in der Konfigurationsdatei festgelegten Datentypen an den entsprechenden Positionen im LCD aus. isdn_rate wird wie folgt aufgerufen:

```
isdn_rate [-ip router-ip] [-port imond-port] [-telmond-port telmond-port]
          [-type hitachi|matrix-orbital|tty] [-config configfilename]
```

Die optionalen Parameter haben folgende Bedeutung:

-ip router-ip Mit `-ip` wird die Adresse des Routers bestimmt, dessen IMOND die Daten liefern soll. Wird der Parameter weggelassen, wird 127.0.0.1 (localhost) verwendet. Es ist möglich, statt der Adresse auch den Namen anzugeben.

-port imond-port `-port` bestimmt den Port auf dem der IMOND die Daten liefert. Standardmässig wird 5000 verwendet.

-telmond-port telmond-port `-telmond-port` bestimmt den Port auf dem der TELMOND die Daten liefert. 5001 ist hier der Defaultwert.

-type hitachi|matrix-orbital|tty `-type` legt den Typ des Displays fest. `hitachi` selektiert Displays mit HD44780 kompatible Anzeigen. `matrix-orbital` entsprechend für Matrix-Orbital-Displays. `tty` ermöglicht die Ausgabe auf der Konsole. WICHTIG: Alle Ausgaben des isdn_rate erfolgen immer auf stdout. Eine Ausgabeumleitung auf die entsprechende Schnittstelle ist also notwendig. tty ist der Standardwert.

-config configfilename `-config` bestimmt den Pfad und den Namen der lcd.conf-Datei. Diese wird vom rc410.lcd-Script erstellt. Der Normalwert ist /var/run/lcd.conf.

isdn_rate gibt es auch in einer Version die unter Windows lauffähig ist. Hierzu muss die Datei /var/run/lcd.conf entweder nach dem Start des Routers von diesem in das isdn_rate-Verzeichnis kopiert werden oder sie muss von Hand erstellt werden. Der Aufruf könnte dann wie folgt aussehen:

```
isdn_rate -ip fli4l -config lcd.conf
```

4.13.4. Anschlußbelegung eines LCD-Moduls am Parallelport

```
  13 _____________________________ 1 Draufsicht auf den
     \ o o o o o o o o o o o o o /   Parallelport, Rück-
      \ o o o o o o o o o o o o /    seite PC
    25 ------------------------- 14
```

Der Anschluß eines LCD-Moduls an den Router wird folgendermaßen aufgetrennt:

Parallelport-Pin	Beschreibung	LCD-Modul	LCD-Pin	
18-25	GND		--|	
	GND		1	--|- Brücke
	R/W		5	--|
	+5V		2	
1	STROBE	EN(1)	6	
2	D0	D0	7	
3	D1	D1	8	
4	D2	D2	9	
5	D3	D3	10	
6	D4	D4	11	
7	D5	D5	12	
8	D6	D6	13	
9	D7	D7	14	
14	Autofeed	RS	4	
17	Select In	EN(2)	?	(für LCDs mit 2 Controller)
Bei Display mit Hintergrundbeleuchtung:				
		HG+	15	(mit Vorwiderstand ca. 20Ohm)
		GND	16	

An Pin 3 kann der Abgriff eines >= 20kOhm Potis zwischen +5V und GND geschaltet werden. Damit kann der Kontrast des Displays reguliert werden. Bei meinem Display (Conrad) liegt Pin 3 direkt an Masse und man kann alles einwandfrei erkennen.

```
+5V ---+
       /
       \ <--+
       /    |
       \    |
GND ---+    +--- VL (Pin 3 - driver input)
```

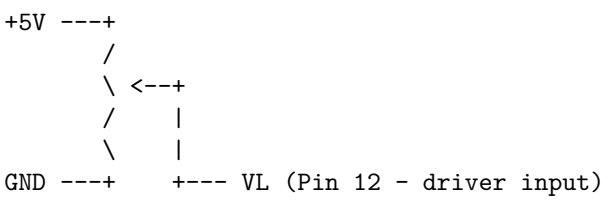
4.13.5. Anschluß eines 4x40 Displays

Da sich der Anschluß eines 4x40 Displays stark von anderen Displays unterscheidet, hier ein Beispiel (Conrad - NLC-40x4x05):

Parallelport-Pin	Beschreibung	LCD-Modul	LCD-Pin	
18-25			--|	
	GND		13	--|- Brücke
	R/W		10	--|

	+5V		14
1	STROBE	EU (Enable-Upper)	9
2	D0	D0	8
3	D1	D1	7
4	D2	D2	6
5	D3	D3	5
6	D4	D4	4
7	D5	D5	3
8	D6	D6	2
9	D7	D7	1
14	Autofeed	RS	11
17	Select In	ED (Enable-Down)	15

An Pin 12 kann der Abgriff eines >= 20kOhm Potis zwischen +5V und GND geschaltet werden. Damit kann der Kontrast des Displays reguliert werden. Es kann aber auch reichen, Pin 12 direkt an Masse zu legen um alles einwandfrei erkennen zu können.



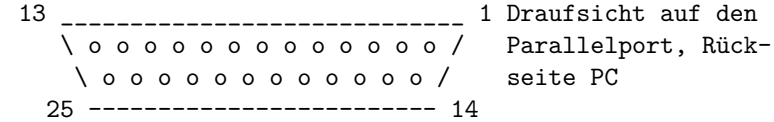
- Die Leitung ED wird an Pin 17 des parallelen Ports angeschlossen.
- Das Display wird in der lcd.txt als 2x40 Display definiert.
- Bei den Typendefinitionen für isdn_rate wird aber die 4x40 als Zeilen-/Spaltengröße angesehen.

Leider gibt es keinen Standard, was die Pinbelegung des Parallelports auf dem Motherboard betrifft. Für die interne Verwendung von LCD-Modulen muß man also die Anschlüsse anhand des zum Motherboard mitgelieferten Slotblechadapters durchmessen.

Die erforderliche Spannungsversorgung kann man leider nicht dem Parallelport entnehmen, da die Stromaufnahme eines LCD-Modules zu hoch ist. Geeignet dafür sind die Anschlüsse für Maus (PS/2), Tastatur (DIN, PS/2), Gameport, USB oder ein freier Anschluß vom PC-Netzteil. Da einige Soundkartenhersteller am Gameport spezielle Signale generieren, kann keine Garantie übernommen, daß es in jeder Kombination funktioniert. Daher gilt hier: Immer vorher messen!

4.13.6. Winamp-Verdrahtung eines LCD-Moduls

Es existieren verschiedene Verdratungsvarianten bei fertig bestückten LCD-Displays von Kernel-Concepts, die normale und die Winamp-Verdrahtung. Neuere Displays kommen mit letzterer Variante.



Der Anschluß eines LCD-Moduls an den Router wird bei der Winamp-Verdrahtung folgendermaßen aufgetrennt:

```
Parallelport-Pin   Beschreibung   LCD-Modul    LCD-Pin
        18-25      GND                        1
           14      Autofeed       R/W         5
                   +5V                        2
            1      STROBE         EN(1)       6
            2      D0             D0          7
            3      D1             D1          8
            4      D2             D2          9
            5      D3             D3         10
            6      D4             D4         11
            7      D5             D5         12
            8      D6             D6         13
            9      D7             D7         14
           16      Init           RS          4

  Bei Display mit Hintergrundbeleuchtung:
                   +5V            HG+        15
                                  GND        16 (mit Regelwiderstand 100Ohm)
```

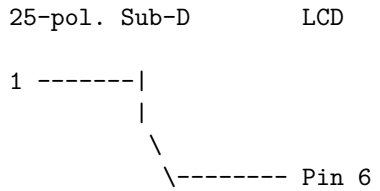
An Pin 3 kann der Abgriff eines >= 10kOhm Potis zwischen +5V und GND geschaltet werden. Damit kann der Kontrast des Displays reguliert werden.

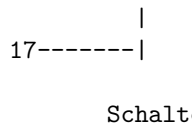
```
+5V ---+
       /
       \ <--+
       /    |
       \    |
GND ---+    +--- VL (Pin 3 - driver input)
```

4.13.7. Tips und Tricks - Zusammengefasst aus Beiträgen von Robert Resch

Anschluss von 2 Displays Mit Hilfe des 2. EN Signals ist es möglich, 2 baugleiche Displays parallel zu betreiben. Dabei wird an Pin 6 des einen Displays Pin 1 (EN1) des Par-Ports angeschlossen und an Pin 6 des 2. Display wird Pin 17 (EN2) angeschlossen. Alle anderen Leitungen werden parallel angeschlossen.

Zwei Displayseiten an einem Display Es sind jetzt auch 2 Displayseiten mit folgender Schaltung möglich:





Dabei wird der gemeinsame Anschluss des Wechslers an Pin 6 des Displays angeschlossen. An den beiden anderen Kontakten werden die beiden EN-Leitungen angeschlossen.

Ansteuerung beider Varianten Das EN2-Signal wird generiert, sobald eine Zeile z mit `LCD_LINES` $< z <$ 2*`LCD_LINES` angesprochen wird. Wenn eine Zeilenzahl noch größer verwendet wird, werden beide Displays angesprochen (um die definierbaren Zeichen wie z.B. die Balkenanzeige von isdn_rate) auf beiden Displays anzeigen zu können. Beide Display-Seiten können eigene definierte Zeichen haben.

Bei 4x40 wird also in der <config>/lcd.txt ein 2x40 Display angegeben. Die Zeilen werden aber trotzdem mit 0-3 angesprochen. Die Zeilennummern 4 und 5 werden in diesem Fall an beide Display-Hälften geschickt. Zeile 4 geht an Zeile 0 und 2, Zeile 5 an Zeile 1 und 3.

Bei z.B. 2 parallel geschalteten 4x20 Displays sieht die Ansteuerung folgendermassen aus:

- Zeilen 0-3 werden auf Display 1 dargestellt
- Zeilen 4-7 werden auf Display 2 dargestellt
- Zeilen 8-11 werden auf beiden Displays dargestellt

Überschwingungen Bei einem langen Kabel oder auch bei bestimmten Parallelports können überschwingende Signale Probleme verursachen. Entweder das Kabel kürzen oder, wenn das nicht geht, eine Terminierung machen. Dazu nimmt man pro Datenleitung (10 Stück!) je einen 10kOhm Widerstand gegen die +5V. Das sollte die Schwinger stabilisieren.

4.13.8. Danksagung

Dank geht an:

- Nils Färber (E-Mail: `nils@kernelconcepts.de`) für den Treiber
- Jürgen Bauer (E-Mail: `jb@idowa.net`) für die erste Version von isdn_rate
- Frank Meyer (E-Mail: `frank@fli4l.de`) für das imond Interface und Fli4l :)
- Ulf Lanz (E-Mail: `u.lanz@odn.de`) für die variable Ausgabe von isdn_rate
- Robert Resch (E-Mail: `rresch@gmx.de`) für Bugfixes und Erweiterungen des lcd-Treibers
- Stefan Krister (E-Mail: `Stefan.Krister@keimfarben.de`) für die erste Überarbeitung der Dokumentation
- Nicole Hornung (E-Mail: `fli4l@xerotech.de`) für ein paar Verbesserungen an der Doku
- Gerrit Lammert (E-Mail: `gerrit@fli4l.de`) für das Umsetzen der Text-Doku in die HTML-Doku

Für Fragen, Anregungen Kritik usw.:
Schreibt eine E-Mail an Gernot Miksch E-Mail: `ibgm@gmx.de`

4.14. LPDSRV - Berkeley lpd Unterstützung

4.14.1. LPDSRV - Druckerserver für lpr/lpd-Protokoll

Den neuen fli4l-Versionen liegt statt des altbekannten LPD ein anderes Programm bei - der LPDSRV. Das ist ein Programm von Steve Flynn und Simon Byrnand. Mit

```
OPT_LPDSRV='yes'
```

kann man fli4l auch als Druckerserver verwenden.
Die wichtigsten Gründe für den Einsatz von LPDSRV statt LPD sind:

- Der LPDSRV ist sehr klein und benötigt deswegen weniger Platz auf der Diskette.
- Er kommt mit sehr wenig Arbeitsspeicher aus, der auch nicht wie beim LPD von vornherein nur für das Drucken reserviert werden muß, da Druckaufträge nicht noch einmal gespoolt werden. Damit kann man auch noch einen 386-er mit 8 MB Arbeitsspeicher und Diskettenaufwerk als reinen Druckserver verwenden.

Leider ergeben sich durch diesen Austausch auch ein paar Einschränkungen:

- Der LPDSRV arbeitet nicht mit allen Druckern problemlos zusammen. Hinweise dazu finden sich unter “Fehlersuche und Beseitigung”.
- Bei Verwendung von LPDSRV kann außerdem nicht über Samba gedruckt werden! Um von Windows 95/98/Me-Clients aus über den LPDSRV zu drucken, muß zwingend ein LPR-Client auf dem Windows-Rechner installiert werden.
- Konfigurierte Drucker tauchen auch bei Installation von Samba und NMBD nicht in der Netzwerkumgebung auf! Wer diese Funktionalität benötigt, muß LPDSRV deaktivieren und (wenn für diese fli4l-Version verfügbar) das optionale Samba-Paket mit SMBD, NMBD und LPD installieren und konfigurieren.
- Der LPDSRV kann insgesamt nur 10 Queues bedienen. Für jeden an fli4l angeschlossenen Drucker werden 2 Warteschlangen erzeugt, eine normale RAW-Queue und eine RAW-Queue mit Seitenvorschub. Damit werden bei Maximal-Ausbau von 3 fli4l-Druckern aber nur 6 Queues benötigt.

Voraussetzungen für ein normales Funktionieren des LPDSRV sind:

- fli4l ist korrekt im Netzwerk eingerichtet.
- die Netzwerke, deren Rechner auf die Drucker an fli4l drucken sollen, sind mittels

  ```
  LPDSRV_NETWORK_N
  LPDSRV_NETWORK_x
  ```

  korrekt konfiguriert worden.
- LPDSRV ist mittels

  ```
  OPT_LPDSRV='yes'
  ```

  aktiviert ;o)

Standard-Einstellung: `OPT_LPDSRV='no'`

4.14.2. Konfiguration

LPDSRV_NETWORK_N Über `LPDSRV_NETWORK_N` wird die Anzahl der Netzwerke bestimmt, von denen aus gedruckt werden kann. Bei einem Netz wird hier

```
LPDSRV_NETWORK_N='1'
```

eingestellt, bei mehreren Netzen ist der Inhalt der Variable zu erhöhen, also z.B.

```
LPDSRV_NETWORK_N='2'
```

Weiterhin müssen dann auch die korrespondierenden Einstellungen *LPDSRV_NETWORK_1* und *LPDSRV_NETWORK_2* vorhanden und mit sinnvollen Werten gefüllt sein.

Standard-Einstellung: `LPDSRV_NETWORK_N='1'`

LPDSRV_NETWORK_x Mit `LPDSRV_NETWORK_x` wird das x'te Netzwerk eingestellt. Bei 2 Netzen müssen 2 Einträge mit den entsprechenden Netzen vorhanden sein, z.B.

```
LPDSRV_NETWORK_1='192.168.6'
LPDSRV_NETWORK_2='10.0.0'
```

Es dürfen in diesem Fall alle Rechner der Netze 192.168.6.0 und 10.0.0.0 drucken. Hierbei ist darauf zu achten, daß nicht '192.168.6.0' oder '10.0.0.0' angegeben und daß nicht auf einem Punkt geendet wird!

Soll der Zugriff von Subnetz 192.168.6.0 und 192.168.7.0 möglich sein, kann man sich die Konfiguration vereinfachen und nur ein Netz definieren, welches diese beiden Subnetze umfasst:

```
LPDSRV_NETWORK_1='192.168'
```

Damit können aber auch Subnetz 192.168.0.0, 192.168.1.0 usw. drucken!

Standard-Einstellung: `LPDSRV_NETWORK_1='192.168.6'`

OPT_LPDSRV_PARPORT Wenn auf Druckern gedruckt werden soll, welche an parallelen Schnittstellen von fli4l hängen, dann ist `OPT_LPDSRV_PARPORT` so zu aktivieren:

```
OPT_LPDSRV_PARPORT='yes'
```

Wichtig: *Bisher wurden nur parallele Schnittstellen auf dem Mainboard oder auf ISA-Schnittstellenkarten unterstützt. PCI-Karten mit parallelen Schnittstellen konnten nicht verwendet werden. Diese Version hier erlaubt auch die Konfiguration von parallelen Schnittstellen auf bestimmten PCI-Karten mit NETMOS-Chips. Hier zu muss man sich mittels*

```
cat /proc/pci
```

die erkannten PCI-Geräte anzeigen lassen. Hier sucht man das Gerät mit der passenden Vendor-ID und Device-ID und wählt als io-Adresse den oder die folgenden Einträge aus:

- *Nm9705CV (Vendor id=9710, Device id=9705, Port1 1. Eintrag)*
- *Nm9735CV (Vendor id=9710, Device id=9735, Port1 3. Eintrag)*
- *Nm9805CV (Vendor id=9710, Device id=9805, Port1 1. Eintrag)*
- *Nm9715CV (Vendor id=9710, Device id=9815, Port1 1. Eintrag, Port2 3. Eintrag)*
- *Nm9835CV (Vendor id=9710, Device id=9835, Port1 3. Eintrag)*
- *Nm9755CV (Vendor id=9710, Device id=9855, Port1 1. Eintrag, Port2 3. Eintrag)*

Die Konfigurationsmöglichkeit wurde eingebaut, ohne entsprechende Hardware zum Testen zur Verfügung zu haben. Daher ist das als experimentelles Feature zu betrachten. Bei Fehlern bitte ausführliche Informationen in die Newsgroup posten!

Standard-Einstellung: `LPDSRV_PARPORT`='no'

LPDSRV_PARPORT_N Über `LPDSRV_PARPORT_N` wird die Anzahl der zu benutzenden Druckerports eingestellt. Bei einem Drucker an der ersten in der lpdsrv.txt konfigurierten parallelen Schnittstelle ist

```
LPDSRV_PARPORT_N='1'
```

einzutragen. Bei 2 Druckerports ist `LPDSRV_PARPORT_N` zu inkrementieren, also

```
LPDSRV_PARPORT_N='2'
```

Weiterhin müssen dann auch die korrespondierenden Einstellungen *LPDSRV_PARPORT_1_IO*, *LPDSRV_PARPORT_1_IRQ*, *LPDSRV_PARPORT_1_DMA* und *LPDSRV_PARPORT_2_IO*, *LPDSRV_PARPORT_2_IRQ*, *LPDSRV_PARPORT_2_DMA* vorhanden sein.

Standard-Einstellung: `LPDSRV_PARPORT_N`='1'

LPDSRV_PARPORT_x_IO Mit `LPDSRV_PARPORT_x_IO` wird der x'te parallele Druckerport eingestellt. Bei 2 Druckern an 2 parallelen Schnittstellen von fli4l müssen 2 Einträge mit den möglichen Werten

- 0x3bc
- 0x378 oder
- 0x278

existieren, also z.B.

```
LPDSRV_PARPORT_1_IO='0x378'
LPDSRV_PARPORT_1_IRQ='no'
LPDSRV_PARPORT_1_DMA='no'
LPDSRV_PARPORT_2='0x278'
LPDSRV_PARPORT_2_IRQ='no'
LPDSRV_PARPORT_2_DMA='no'
```

Wie die Portdefinitionen bei Netmos-PCI-Parallelportkarten aussehen müssen, könnt Ihr unter OPT_LPDSRV_PARPORT nachlesen. Man sollte sich vor der Konfiguration unbedingt vergewissern, auf welche io-Adressen die eingebauten Schnittstellen eingestellt sind, da der Druck sonst nicht funktioniert. Die io-Adressen kann man entweder im BIOS seines Rechners einstellen oder sie sind bei sehr alten Rechnern nicht konfigurierbar, werden aber beim Booten angezeigt. Zusätzlich verbaute Ports lassen sich meist über Jumper auf der io-Karte einstellen und werden in der (hoffentlich noch vorhandenen Dokumentation) zur Einstellung der Druckerports beschrieben. Außerdem ist darauf zu achten, daß bei OPT_-LCD='yes' die hier eingestellten Adressen in der lpdsrv.txt nicht mit der dort eingestellten io-Adresse in LCD_ADDRESS kollidieren. Dieser Konflikt verhindert eine Diskettenerstellung!

Standard-Einstellung: LPDSRV_PARPORT_1_IO='0x378'

LPDSRV_PARPORT_x_IRQ Über LPDSRV_PARPORT_x_IRQ wird eingestellt, ob im Interruptbetrieb gedruckt werden soll, was den Prozessor entlastet. Dazu muss bei Schnittstellen auf dem Mainboard oder auf ISA-Karten aber im Rechnerbios oder per Jumperbelegung in jedem Fall der ECP/EPP-Modus konfiguriert werden. Der Interruptbetrieb wird so aktiviert:

```
LPDSRV_PARPORT_1_IRQ='yes'
```

Will man diesen Modus nicht nutzen, so ist

```
LPDSRV_PARPORT_1_IRQ='no'
```

zu setzen und bei Schnittstellen auf dem Mainboard oder auf ISA-Karten im Rechnerbios oder per Jumperbelegung in jedem Fall der Normal- oder SPP-Modus zu konfigurieren. Wenn etwas nicht funktioniert, sollte auf jeden Fall erst einmal mit

```
LPDSRV_PARPORT_1_IRQ='no'
```

getestet werden.

Standard-Einstellung: LPDSRV_PARPORT_1_IRQ='no'

LPDSRV_PARPORT_x_DMA Über LPDSRV_PARPORT_x_DMA wird eingestellt, ob im DMA-Betrieb gedruckt werden soll, was den Prozessor entlastet. Dazu muss bei Schnittstellen auf dem Mainboard oder auf ISA-Karten aber im Rechnerbios oder per Jumperbelegung in jedem Fall der ECP/EPP-Modus konfiguriert werden. Der Interruptbetrieb wird so aktiviert:

```
LPDSRV_PARPORT_1_DMA='yes'
```

Voraussetzung dafür ist

```
LPDSRV_PARPORT_1_IRQ='yes'
```

Will man diesen Modus nicht nutzen, so ist

```
LPDSRV_PARPORT_1_DMA='no'
```

zu setzen und bei Schnittstellen auf dem Mainboard oder auf ISA-Karten im Rechnerbios oder per Jumperbelegung in jedem Fall der Normal- oder SPP-Modus zu konfigurieren. Wenn etwas nicht funktioniert, sollte auf jeden Fall erst einmal mit

```
LPDSRV_PARPORT_1_DMA='no'
```

getestet werden.

Standard-Einstellung: `LPDSRV_PARPORT_1_DMA='no'`

OPT_LPDSRV_USBPORT Wenn auf Druckern gedruckt werden soll, welche an USB-Schnittstellen von fli4l hängen, dann ist `OPT_LPDSRV_USBPORT` so zu aktivieren:

```
OPT_LPDSRV_USBPORT='yes'
```

Ausserdem muss die generelle Unterstützung für USB-Drucker im separaten Paket USB eingeschaltet werden. Siehe dazu USB - Support für USB-Geräte (Seite 247). Je nach Treiber sieht das so aus:

```
OPT_USB='yes'
USB_LOWLEVEL='uhci'
USB_EXTRA_DRIVER_N='1'
USB_EXTRA_DRIVER_1='printer'
USB_EXTRA_DRIVER_1_PARAM=''
```

oder so:

```
OPT_USB='yes'
USB_LOWLEVEL='ohci'
USB_EXTRA_DRIVER_N='1'
USB_EXTRA_DRIVER_1='printer'
USB_EXTRA_DRIVER_1_PARAM=''
```

Wichtig: *Die Konfigurationsmöglichkeit für USB-Drucker wurde eingebaut, ohne entsprechende Hardware zum Testen zur Verfügung zu haben. Daher ist das als experimentelles Feature zu betrachten. Bei Fehlern bitte ausführliche Informationen in die Newsgroup posten! Viele USB-Drucker sind GDI-Drucker. GDI-Drucker, deren Treiber eine bidirektionale Kommunikation benötigt, können nicht angesprochen werden. Das ist bei den meisten GDI-Druckern der Fall. Ich werde nur Fragen zu Problemen mit USB-Druckern beantworten, aus denen hervorgeht, dass Ihr ausgeschlossen habt, dass der betroffene Drucker ein GDI-Drucker ist bzw. als solcher eine bidirektionale Kommunikation benötigt!*

Standard-Einstellung: `LPDSRV_USBPORT='no'`

LPDSRV_USBPORT_N Über `LPDSRV_USBPORT_N` wird die Anzahl der zu benutzenden USB-Druckerports eingestellt. Bei einem Drucker ist

```
LPDSRV_USBPORT_N='1'
```

einzutragen. Bei 2 USB-Druckerports ist `LPDSRV_USBPORT_N` zu inkrementieren, also

```
LPDSRV_USBPORT_N='2'
```

Wichtig: *Wenn mehr als ein USB-Drucker verwendet wird, ist darauf zu achten, dass durch die Reihenfolge des Einschaltens der Drucker bestimmt wird, welcher Drucker der erste und welcher Drucker der zweite USB-Drucker wird. Der zweite USB-Drucker wird ausserdem automatisch zum ersten Drucker, wenn der erste USB-Drucker überhaupt nicht eingeschaltet wird. Wenn es sich um unterschiedliche Drucker-Modelle handelt, die unterschiedliche Treiber auf dem Client benötigen, kann es dadurch passieren, dass auf dem gewählten Drucker nur Zeichensalat ausgegeben wird, da der Druckjob in der Druckersprache des anderen Druckers formatiert wurde.*

Standard-Einstellung: `LPDSRV_USBPORT_N='1'`

OPT_LPDSRV_REMOTE Wenn auf Printserver gedruckt werden soll, die es erlauben, Daten per ftp oder netcat an sie zu schicken, dann ist `OPT_LPDSRV_REMOTE` so zu aktivieren:

```
OPT_LPDSRV_REMOTE='yes'
```

Printserver, welche über das lpd-Protokoll drucken, können nicht angesteuert werden! Ob Euer Printserver zur unterstützten Kategorie gehört, entnehmt Ihr bitte dem mitgelieferten Handbuch oder der Webseite des Herstellers. Eine unvollständige Übersicht findet Ihr unter

http://www2.cruzio.com/~jeffl/sco/lp/printservers.htm

Ich habe nicht die Zeit, Euch diese Informationen herauszusuchen, recherchiert also bitte selbst.

LPDSRV_REMOTE_N Über `LPDSRV_REMOTE_N` wird die Anzahl der zu benutzenden externen Printserveranschlüsse eingestellt. Dabei ist zu beachten, dass manche Printserver mehrere Anschlüsse besitzen. Möchte man 2 Printserver mit jeweils 2 Anschlüssen ansteuern, ist also

```
LPDSRV_REMOTE_N='4'
```

einzustellen. Bei einem Drucker an einem Anschluss eines externen Printservers konfiguriert man

```
LPDSRV_REMOTE_N='1'
```

Bei 2 Druckern an Anschlüssen ist `LPDSRV_REMOTE_N` zu erhöhen, also

```
LPDSRV_REMOTE_N='2'
```

Weiterhin müssen dann auch die korrespondierenden Einstellungen *LPDSRV_REMOTE_1_IP*, *LPDSRV_REMOTE_2_IP*, *LPDSRV_REMOTE_1_PORTNR* und *LPDSRV_REMOTE_2_PORTNR* vorhanden und korrekt konfiguriert sein.

Standard-Einstellung: LPDSRV_REMOTE_N='0'

LPDSRV_REMOTE_x_IP Mit LPDSRV_REMOTE_x_IP wird die IP-Adresse des x'ten Printserveranschlusses eingestellt. Hat man einen Printserver mit 2 benutzten Anschlüssen, so wird

```
LPDSRV_REMOTE_N='2'
```

und bei *LPDSRV_REMOTE_1_IP* und *LPDSRV_REMOTE_2_IP* die IP-Adresse dieses Printservers eingestellt (also 2 mal dieselbe IP).

Standard-Einstellung: LPDSRV_REMOTE_1_IP='192.168.6.99'

LPDSRV_REMOTE_x_PORTNR Mit LPDSRV_REMOTE_x_PORTNR wird die Portnummer des x'ten Printserver- anschlusses eingestellt. Die Information darüber, welche Portnummern Euer Printserver an welchem Anschluss standardmässig verwendet, entnehmt Ihr bitte dem mitgelieferten Handbuch oder der Webseite des Herstellers, denn ich habe nicht die Zeit, diese Informationen für Euch herauszusuchen. Eine unvollständige Übersicht findet Ihr unter

http://www2.cruzio.com/~jeffl/sco/lp/printservers.htm

Achtet ausserdem darauf, dass bei manchen Printservern die Portnummern konfigurierbar sind!

Standard-Einstellung: LPDSRV_REMOTE_1_PORTNR='9100'

Für jeden an lokalen parallelen Schnittstellen an fli4l hängenden Drucker werden durch das Startscript des LPDSRV 2 Druckerqueues angelegt, die wie folgt definiert sind:

Erster Drucker

- pr1 (normale RAW-Queue)
- prf1 (RAW-Queue mit Seitenvorschub)

Zweiter Drucker

- pr2 (normale RAW-Queue)
- prf2 (RAW-Queue mit Seitenvorschub)

usw.

Für jeden an lokalen USB-Schnittstellen an fli4l hängenden Drucker werden durch das Startscript des LPDSRV 2 Druckerqueues angelegt, die wie folgt definiert sind:

Erster Drucker

- usbpr1 (normale RAW-Queue)

- usbprf1 (RAW-Queue mit Seitenvorschub)

Zweiter Drucker

- usbpr2 (normale RAW-Queue)
- usbprf2 (RAW-Queue mit Seitenvorschub)

usw.

Für jeden konfigurierten externen Printserveranschluss wird durch das Startscript des LPDSRV eine Druckerqueue angelegt, die wie folgt definiert ist:

Erster Anschluss

- repr1 (normale RAW-Queue)

Zweiter Anschluss

- repr2 (normale RAW-Queue)

usw.

4.14.3. Einrichtung eines Linux-LPR-Clients

Auf einem Linux-Rechner kann der fli4l-Netzwerkdrucker in der Datei /etc/printcap eingetragen werden.

Beispiel (Name des Druckers: "drucker"):

```
drucker:\
            :lp=:\
            :rm=fli4l:\
            :rp=pr1:\
            :sd=/var/spool/lpd/drucker:\
            :sh:mx#0:
```

Dabei wird mit "rm=fli4l" der Rechnername des fli4l-Routers angegeben. Dieser ist gegebenenfalls anzupassen. Soll die Linux-Drucker-Queue anders heißen, ist "drucker" ebenfalls anzupassen.

Der Remote-Warteschlangenname in "rp=pr1" lautet wie folgt:

:rp=pr1: für den ersten an fli4l parallel angeschlossenen Drucker

:rp=pr2: für den zweiten an fli4l parallel angeschlossenen Drucker

:rp=prf1: bzw. :rp=prf2: für die parallelen Formfeed-Queues

:rp=usbpr1: für den ersten an fli4l per USB angeschlossenen Drucker

:rp=usbpr2: für den zweiten an fli4l per USB angeschlossenen Drucker

:rp=usbprf1: bzw. :rp=usbprf2: für die USB-Formfeed-Queues

:rp=repr1: bzw. :rp=repr2: für die konfigurierten Remote- Printerserveranschlüsse

Wichtig: *Nach Einfügen des Eintrages in der Datei /etc/printcap muss das Verzeichnis /var/spool/lpd/drucker mit dem mkdir-Kommando selbst eingerichtet werden.*

Mit dem Kommando "lpr -P drucker DATEINAME" kann man nun Dateien vom Linux-Rechner über fli4l ausdrucken.

Viele neuere Distributionen verwenden alternative Drucksysteme und eigene Konfigurationstools, bei denen die oben beschriebene Konfiguration misslingt. Peter Schöne hat aus diesem Grund eine Beschreibung für die in Deutschland weit verbreitete Distribution SuSE (in der Version 8.1) beigesteuert:

Unter Suse 8.1 mit dem Standarddrucksystem CUPS ist die Einrichtung sehr viel komfortabler.

Unter YAST2 wählt man aus der Rubrik Hardware die Druckerkonfiguration. Falls man die lokalen Drucker bereits eingerichtet hat, kann man die automatische Erkennung getrost überspringen ;-) Im Fenster "Druckereinrichtung" wählt man den Button "Konfigurieren...", im darauffolgenden die Rubrik "Mehr Anschlussmöglichkeiten zeigen..." und bestätigt durch "Weiter". Es werden nun verschiedene Druckertypen angezeigt. Da es sich hier um ein LPD-kompatibles Paket handelt, wählt man gleich den ersten Eintrag "LPD-Vorfilter- und -Weiterleitungs-Warteschlange". Nach einer erneuten Bestätigung mit "Weiter" gelangt man zur eigentlichen Konfiguration: Man kann sich hier, wenn man unsicher ist, den Namen des Routers durch den Button "Lookup"-"LPD-Servers" automatisch eintragen lassen, oder die IP-Adresse des Routers direkt eingeben. In das zweite Feld kommt der Name der Druckerwarteschlange. Dabei ist beim ersten parallel an fli4l angeschlossenen Drucker "pr1", beim zweiten "pr2", beim dritten "pr3" einzutragen, bzw. für die Warteschlangen mit Seitenvorschub "prf1", "prf2" und "prf3". Für die per USB an fli4l angeschlossenen Drucker ist "usbpr1", "usbpr2" usw. einzutragen, bzw. für die Warteschlangen mit Seitenvorschub "usbprf1", "usbprf2" usw. Für konfigurierten Remote-Drucker ist "repr1", "repr2" usw. einzutragen. Ein Klick auf den Button "Entfernten LPD-Zugang testen" zeigt, ob die Einstellungen korrekt sind. Wird dies bestätigt, kann man im Dialog mit "Weiter" fortfahren. Im folgenden Fenster wird ein Namen, unter dem der Drucker aus Anwendungen heraus druckt, vergeben. Die Felder "Beschreibung des Druckers" und "Standort des Druckers" bleiben leer. Es geht noch "Weiter" ... Man wählt nun den am Router angeschlossenen Drucker aus, bestätigt die Auswahl, entscheidet sich für die richtigen Treiber und schließt die gesamte Konfiguration mit einem Klick auf den Button "Beenden" und einer Bestätingung mit "Ja" ab. Der Drucker ist nunmehr vollständig eigerichtet und sollte aus den meisten Anwendungen heraus drucken.

Für eine Installation unter StarOffice oder OpenOffice führt man das mitgelieferte Programm "Drucker Verwaltung" aus und installiert den neu hinzugekommenen Drucker als "Generic Printer".

Wichtig: *Wenn bei Verwendung des Drucksystems Cups Probleme auftreten, kann das daran liegen, dass lpdsrv Anfragen von bestimmten reservierten Ports erwartet, diese aber von neueren cups-Versionen nicht genutzt werden. In diesem Fall muss die Datei printer.conf auf dem Linux-Client editiert werden. Es ist an den Eintrag für den Warteschlangennamen "?reserve=yes" anzuhängen. Ein vollständiger Eintrag könnte dann (natürlich abhängig von Euren Gegebenheiten) so aussehen:*

```
lpd://192.168.0.1/pr1?reserve=yes
```

4.14.4. Einrichtung eines Windows NT 4.0/2000-LPR-Clients

Unter Windows NT 4.0/2000 ist für den Zugriff auf den LPDSRV von fli4l die Installation der Druckdienste für TCP/IP / Unix notwendig, da beim Drucken über den Standard-TCP/IP-Port von Windows ungeeignete Ports benutzt werden. Es ist aber keine weitere Zusatzsoftware erforderlich, wie bei Windows 95/98/Me.

Die Druckdienste für TCP/IP können bei NT 4.0 über

```
Start
   Einstellungen
      Systemsteuerung
         Netzwerk
            Hinzufügen
               Dienste
                  Microsoft TCP/IP Druckdienst
```

hinzugefügt werden.

Die Druckdienste für Unix können bei Windows 2000 über

```
Start
   Einstellungen
      Systemsteuerung
         Software
            Windows-Komponenten hinzufügen
               Weitere Datei- und Druckdienste für das Netzwerk
                  Details
                     Druckdienste für UNIX
```

hinzugefügt werden.

Damit ist ein neuer Druckerport mit dem Namen “LPR Port” verfügbar. Nun richtet man mit dem Druckerassistenten unter Windows NT 4.0/2000 einen neuen Drucker mit dem Treiber des an fli4l hängenden Druckers ein. Dazu geht man auf

```
Start
   Einstellungen
      Drucker
```

und macht einen Doppelklick auf “Neuer Drucker”. Hier bestätigt man die Einleitung mit “Weiter”, wählt “Lokaler Drucker” aus, deaktiviert “Automatische Druckererkennung und Installation von Plug & Play-Druckern” und bestätigt mit “Weiter”. Unter “Druckeranschluß auswählen” aktiviert man “Einen neuen Anschluß erstellen” und wählt unter “Typ” den oben erstellten “LPR Port”. Nachdem man diese Einstellungen mit dem Drücken von “Weiter” bestätigt hat, trägt man in das Feld “Name oder Adresse des Servers für LPD” die richtige IP-Adresse des fli4l-Rechners ein und schreibt in das Feld “Name des Druckers oder der Druckerwarteschlange auf dem Server” den Namen der richtigen Druckerqueue. Dabei ist beim ersten parallel an fli4l angeschlossenen Drucker “pr1”, beim zweiten “pr2” und beim dritten “pr3” einzutragen, bzw. für die Warteschlangen mit Seitenvorschub “prf1”, “prf2” und “prf3”. Für per USB an fli4l angeschlossene Drucker heissen die Druckerqueues “usbpr1”, “usbpr2” usw., bzw. für die

Warteschlangen mit Seitenvorschub “usbprf1”, “usbprf2” und “usbprf3”. Für die Anschlüsse der von fli4l extern angesteuerten Printserver setzt man “repr1”, “repr2” usw.

Auf dem nächsten Konfigurationsbildschirm wählt man auf der linken Seite den Hersteller des an fli4l hängenden Druckers und auf der rechten Seite den entsprechenden Typ aus und bestätigt abermals mit “Weiter”. Im Feld “Druckername” kann man nun einen Namen für den Drucker festlegen. Unter Druckerfreigabe wählt man “Diesen Drucker nicht freigeben”, da der Drucker am fli4l-Rechner freigegeben ist. Nach dem Klick auf “Weiter” verneint man die Frage, ob eine Testseite gedruckt werden soll, da noch nicht alle Einstellungen vorgenommen worden sind und bestätigt wieder mit “Weiter”.

Nun erscheint ein Fenster mit der Zusammenfassung der bisherigen Konfiguration. Wenn alles korrekt eingegeben wurde, drückt man “Fertig stellen”.

Nach dem Kopieren des Druckertreibers erscheint ein neues Icon für diesen Drucker im Druckerordner.

Das Icon für den fli4l-Drucker klickt man mit der rechten Maustaste an und wählt aus dem Kontextmenü “Eigenschaften”. Auf der Lasche “Anschlüsse” deaktiviert man “Bidirektionale Unterstützung aktivieren”.

Auf der Lasche “Erweitert” betätigt man die Schaltfläche “Druckprozessor” und stellt unter “Druckprozessor” “WinPrint”, unter “Standarddatentyp” “RAW” ein und verläßt diese Dialogbox mit “OK” (bei Windows NT 4.0 ist hier noch ein Häckchen bei “Raw-Datentyp immer spoolen” zu setzen). Wieder auf der Lasche “Erweitert” aktiviert man “Über Spooler drucken, um Druckvorgänge schneller abzuschließen” und “Drucken beginnen, nachdem letzte Seite gespoolt wurde”.

Bei “Erweiterte Druckfunktionen aktivieren” entfernt man den Haken, damit diese Funktionen nicht genutzt werden. Jetzt übernimmt man alle bisher gemachten Einstellungen mit der Schaltfläche “Übernehmen” und verläßt das komplette Konfigurationsfenster über “OK”, da Windows NT 4.0/2000 die Einstellungen sonst nicht korrekt abspeichert.

4.14.5. Einrichtung eines Windows 95/98/Me-LPR-Clients

Unter Windows 95/98/Me ist für den Zugriff auf den LPDSRV von fli4l Zusatzsoftware erforderlich, da Microsoft diesen Versionen im Gegensatz zu Windows NT 4.0/2000 keinen LPR-Client eingebaut hat.

Das Drucken über LPR von diesen Windowsversionen aus wurde mit der ACITS-LPR-Clientsoftware getestet, deren ältere Versionen für den Privatgebrauch und den universitären Bereich kostenfrei waren. Die aktuellen Versionen sind es leider nicht mehr. Die ältere, kostenfreie Version, ist aber noch an einigen Stellen im Internet zu finden (Version 3.4f).

Eine zum Zeitpunkt der Erstellung dieser Dokumentation noch funktionierende Downloadmöglichkeit war
`ftp://ftp.informatik.uni-hamburg.de/pub/os/unix/utils/LPRng/WINDOWS/acitsplr/instlpr.exe`

Unter `http://www.utexas.edu/academic/otl/software/lpr/` findet man nur noch die aktuellen, laufzeitbeschränkten Testversionen.

Es gibt noch andere freie LPR-Clients für Windows, Links dazu finden sich auf der fli4l-Homepage unter `http://www.fli4l.de/sonstiges/links/interessantes/`

Beschrieben wird hier aber nur die Installation und Konfiguration der ACITS-Software.

Zur Installation ist die aus dem Internet geladene Datei `INSTLPR_95`.EXE zu starten. Am Ende der Installation kann man sich die mitgelieferte englische Konfigurationsanleitung anzeigen

lassen - das kann man aber auch lassen, da die Konfiguration hier beschrieben wird ;o)

Zuerst richtet man mit dem Druckerassistenten unter Windows 95/98/Me einen lokalen Drucker mit dem Treiber für den am fli4l-Rechner angeschlossenen Drucker ein. Dazu geht man auf

```
Start
   Einstellungen
      Drucker
```

und macht einen Doppelklick auf “Neuer Drucker”. Hier bestätigt man die Einleitung mit “Weiter”, wählt “Lokaler Drucker” aus, bestätigt mit “Weiter”, wählt auf der linken Seite den Druckerhersteller und auf der rechten Seite den Druckertyp aus und bestätigt abermals mit “Weiter”. Bei der Frage nach dem gewünschten Anschluß für den Drucker markiert man “LPT1: Druckeranschluß” und geht auf “Weiter”.

Auf dem nächsten Konfigurationsbildschirm wählt man einen Namen für den Drucker aus und aktiviert, wenn gewünscht, daß dieser Drucker der Standarddrucker unter Windows werden soll.

Nach dem unvermeidlichen Klick auf “Weiter” verneint man die Frage, ob eine Testseite gedruckt werden soll, da der Drucker ja nicht wirklich am LPT1 des Client-Rechners hängt und der korrekte Anschluß noch nicht konfiguriert wurde. Nach Betätigen von “Fertigstellen” und dem Kopieren der Treiberdateien erscheint ein neues Icon für diesen Drucker im Druckerordner. Das Icon für den fli4l-Drucker klickt man mit der rechten Maustaste an und wählt aus dem Kontextmenü “Eigenschaften”. Auf der Lasche “Details” konfiguriert man den Druckeranschluß über “Anschluß hinzufügen”, wählt “Andere” und darunter “ACITS LPR Remote Printing” und bestätigt mit “OK”. Jetzt öffnen sich die Eigenschaften des Druckeranschlusses des LPR-Clients. Hier trägt man auf der Lasche “Settings” in das Feld “Host name or IP address” die richtige IP-Adresse des fli4l-Rechners ein und schreibt in das Feld “Printer/Queue name” den Namen der richtigen Druckerqueue. Dabei ist beim ersten parallel an fli4l angeschlossenen Drucker “pr1”, beim zweiten “pr2” und beim dritten “pr3” einzutragen, bzw. für die Warteschlangen mit Seitenvorschub “prf1”, “prf2” und “prf3”. Für per USB an fli4l angeschlossene Drucker heissen die Druckerqueues “usbpr1”, “usbpr2” usw., bzw. für die Warteschlangen mit Seitenvorschub “usbprf1”, “usbprf2” und “usbprf3”. Für die Anschlüsse der von fli4l extern angesteuerten Printserver setzt man “repr1”, “repr2” usw. Wenn der Client, an dem man sitzt, in der base.txt mit Name und IP-Adresse richtig konfiguriert wurde, dann erhält man bei Überprüfung der Zugriffsmöglichkeit auf die Druckerwarteschlange über die Schaltfläche “Verify Printer Information” die Meldung “The printer/queue name (Name) is accessable on the remote host (IP)”. Hier steht natürlich bei jedem die unter “Host name or IP address” hinterlegte IP-Adresse des fli4l-Rechners und die entsprechende Warteschlange.

Sollte eine Fehlermeldung erscheinen, sind die Einstellungen für DNS des Client-Rechners oder für den Drucker in der lpdsrv.txt oder für den Warteschlangennamen nicht korrekt.

Auf der Lasche “LPR Options” kann man nun einstellen, ob man eine Trennseite zwischen den einzelnen Druckaufträgen drucken will, um die Drucke dem richtigen Nutzer zuordnen zu können. Diese Option benötigt man nur bei sehr vielen Client-Rechnern. Meist kann man die Option abschalten und nimmt den Haken bei “Enable banner page” heraus. Jetzt bestätigt man die Konfiguration mit “OK”.

Als Nächstes bearbeitet man auf der Lasche Details die “Spool-Einstellungen” und setzen dort “Druckaufträge in Warteschlange stellen (Druckvorgang schneller)” und “Druck nach letzter Seite beginnen”. Unter Datenformat wählt man “RAW” und setzt außerdem “Bidirektionale

Unterstützung deaktivieren". Mit "OK" verläßt man die Spooleinstellungen und aktiviert die Lasche "Allgemein", um über "Testseite drucken" zu sehen, ob die Konfiguration Erfolg hatte.

4.14.6. Einrichtung eines Mac-Clients (MacOSX 10.3.2)

Hier öffnet man in den "Systemeinstellungen" das "Drucker Dienstprogramm" und drückt "Hinzufügen". Hier wird "TCP/IP - Drucker" ausgewählt und als Druckertyp "LPD/LPR". Unter "Druckeradresse" wird die IP-Adresse des Routers eingetragen. Nun ist der "Name der Warteliste" anzugeben. Dabei ist beim ersten parallel an fli4l angeschlossenen Drucker "pr1", beim zweiten "pr2" und beim dritten "pr3" einzutragen, bzw. für die Warteschlangen mit Seitenvorschub "prf1", "prf2" und "prf3". Für per USB an fli4l angeschlossene Drucker heissen die Druckerqueues "usbpr1", "usbpr2" usw., bzw. für die Warteschlangen mit Seitenvorschub "usbprf1", "usbprf2" und "usbprf3". Für die Anschlüsse der von fli4l extern angesteuerten Printserver setzt man "repr1", "repr2" usw. Dann wählt man das Druckermodell aus der Auswahlliste und klickt zum Schluss auf "Hinzufügen".

4.14.7. Fehlersuche und Beseitigung

Fehler: Die letzte Seite des Druckjobs wird nicht zu Ende gedruckt und nicht ausgeworfen.

Lösung: Dieser Fehler tritt vor allem bei verschiedenen HP-Deskjet-Druckern auf. Hier hilft es, die Formfeedtaste zu betätigen, die Formfeedqueue zu verwenden oder noch einen Druckjob abzuschicken. Es besteht auch die Chance, dass die entsprechenden Drucker mit dem LPD aus dem Paket `OPT_SAMBA_LPD` problemlos zusammenarbeiten oder dass ein aktuellerer oder älterer Druckertreiber von der Webseite des Druckerherstellers das Problem beseitigt.

Fehler: Kein Zugriff auf den lpdsrv vom Windows-Client aus.

Lösung:

- Es wurden falsche Warteschlangennamen eingetragen $\Longrightarrow$ korrigieren.
- Die Variable `LPDSRV_NETWORK` wurde falsch gefüllt $\Longrightarrow$ korrigieren.

Fehler: Mit dem Epson Stylus Color ist der Druck nicht möglich.

Lösung: Verwenden des Windows-Druck-Managers statt des Spoolmanagers von Epson. Hierzu ruft man den Spool-Manager mit "EPSPMGR2.EXE" (nicht "epdspl2.exe") aus dem Systemverzeichnis von Windows auf. Hinweis: Aktuelle Druckertreiber- Versionen installieren in der Programm- gruppe "EPSON Drucker" den Spool- Manager automatisch, d.h. in diesem Fall reicht ein Doppelklick auf das entsprechende Symbol. Bei markiertem EPSON -Drucker wechselt man dann in das Setup. In der Registerkarte "Warteschlange einrichten" klickt man auf das Kontrollkästchen "Druckmanager für diesen Port verwenden".

Fehler: Der Ausdruck wird vom Windows-Clients aus ohne Probleme abgeschickt, es wird trotzdem nichts gedruckt.

Lösung:

- Der Drucker ist vielleicht defekt - am besten mitsamt Kabel an einem anderen Rechner ausprobieren.

- Die Schnittstellenkarte ist defekt $\Longrightarrow$ austauschen.
- Die Druckeinstellungen im Treiber wurden nicht nach Dokumentation vorgenommen $\Longrightarrow$ korrigieren.
- Der LPT-Port wird in einem zu schnellen Modus angesprochen $\Longrightarrow$ korrigieren. Im BIOS oder auf der Schnittstellenkarte sollte wenn möglich “SPP” bzw. “Normal” bzw. “Standard” eigestellt werden.
- Eventuell wird ein GDI-Drucker benutzt. Solche Drucker sind nur unter Windows lauffähig. Ob Ihr so einen Drucker habt, verrät das Handbuch bzw. die Webseite des Druckerherstellers. Die Lösung besteht im Austausch des Druckers gegen einen anderen, der auch ohne Windows lauffähig ist.

Fehler: Bei Verwendung des Druckystems Cups auf Linux-Clients wird nicht gedruckt und folgende Meldung steht im Cups Error-Log: “Remote host did not accept control file”

Lösung: Das kann daran liegen, dass lpdsrv Anfragen von bestimmten reservierten Ports erwartet, diese aber von neueren cups-Versionen nicht genutzt werden. In diesem Fall muss die Datei printer.conf auf dem Linux-Client editiert werden. Es ist an den Eintrag für den Warteschlangennamen “?reserve=yes” anzuhängen. Ein vollständiger Eintrag könnte dann (natürlich abhängig von Euren Gegebenheiten) so aussehen:

```
lpd://192.168.0.1/pr1?reserve=yes
```

Fehler: Es hängen 2 USB-Drucker an fli4l. Wenn einer davon ausgeschaltet ist, wird auf dem andere nur noch Zeichensalat ausgegeben.

Lösung: Wenn mehr als ein USB-Drucker verwendet wird, ist darauf zu achten, dass durch die Reihenfolge des Einschaltens der Drucker bestimmt wird, welcher Drucker der erste und welcher Drucker der zweite USB-Drucker wird. Der zweite USB-Drucker wird ausserdem automatisch zum ersten Drucker, wenn der erste USB-Drucker überhaupt nicht eingeschaltet wird. Wenn es sich um unterschiedliche Drucker-Modelle handelt, die unterschiedliche Treiber auf dem Client benötigen, kann es dadurch passieren, dass auf dem gewählten Drucker nur Zeichensalat ausgegeben wird, da der Druckjob in der Druckersprache des anderen Druckers formatiert wurde. Hier hilft es nur, die Drucker in der Reihenfolge anzuschalten, wie sie zum Konfigurationszeitraum gegeben war und immer alle USB-Drucker angeschaltet zu lassen.

4.15. OpenVPN - VPN-Support

Ab Version 2.1.5 ist das OpenVPN Paket fester Bestandteil von fli4l.

Wichtig: *Für die Nutzung von OpenVPN über das Internet, ist in jedem Fall eine Flatrate oder eine volumenbasierte Abrechnung notwendig! Wenn der fli4l-Router permanent eingeschaltet bleibt, wird die Verbindung nicht getrennt, da permanent Daten (wenn auch nur ein paar Bytes alle paar Sekunden) übertragen werden. Mit der Nutzung eines der VPN Pakete und der Verwendung eines VPN Tunnel über das Internet, legt der fli4l-Router nicht mehr auf und es entstehen hohe Kosten, wenn keine Flatrate oder ein volumenbasiertes Abrechnungsmodel benutzt wird! Gleiches gilt natürlich für eine ISDN Wählleitung.*

Neben OpenVPN gibt es in der opt-Datenbank http://www.fli4l.de/download/zusatzpakete/ noch das VPN Paket OPT_PoPToP.

Die Entscheidung für eine VPN Lösung hängt in erster Linie von der Sicherheit und der Funktion der eingesetzten Lösung ab. Aussagen zur Sicherheit der hier angebotenen VPN Lösungen gibt das fl4il Team bewusst nicht ab, verweist dafür auf folgende Webseiten und Berichte:

Linux-Magazin Ausgabe Januar 2004

http://diswww.mit.edu/bloom-picayune/crypto/14238

http://sites.inka.de/bigred/archive/cipe-l/2003-09/msg00263.html

Zur Funktionalität kann das fli4l Team aber eine klare Aussage treffen. In diesem Punkt ist OpenVPN der klare Gewinner gegenüber CIPE und poptop. OpenVPN unterstüzt neben einem Tunnel- und Bridgemodus auch Datenkompression und läuft im Gegensatz zu CIPE wesentlich stabiler auf dem fli4l-Router. Außerdem gibt es für OpenVPN auch eine Windows Version, die ab Windows 2000 eingesetzt werden kann. Einziger Nachteil von OpenVPN ist seine Größe im opt-Archiv gegenüber CIPE und die fehlende OpenVPN Unterstützung für fli4l Version 2.0.x.

4.15.1. OpenVPN - Einführendes Beispiel

Um den Einstieg in die Konfiguration zu erleichtern, vorab ein kleines Beispiel. Es sollen zwei Netze, die beide einen fli4l-Router einsetzen, über das Internet verbunden werden. Dazu wird von OpenVPN auf den zwei fli4l-Routern ein verschlüsselter Tunnel eingerichtet, durch den die Computer aus den enfernten Netzen miteinander kommunizieren können. Dabei spielen die im Bild 4.1 gezeigten Konfigurationsvariablen eine Rolle.

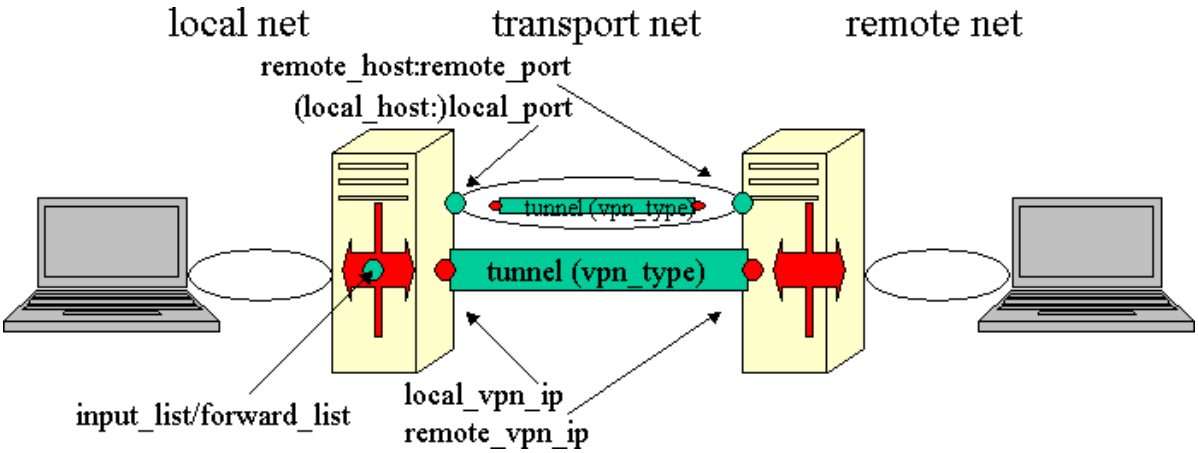


Abbildung 4.1.: VPN-Konfigurationsbeispiel — Tunnel zwischen zwei Routern

local net, remote net repräsentieren die beiden Netze, die über den Tunnel miteinander verbunden werden sollen. Die beiden zu verbindenden Netze müssen in unterschiedlichen TCP/IP Netzen sein und dürfen sich in ihren Netzmasken auch nicht überschneiden. Die Einstellungen von IP_NET_x (Seite 39) in der jeweiligen base.txt Konfigurationsdateien dürfen also nicht gleich sein. Es ist also mit einem VPN Tunnel nicht möglich zwei Netze miteinander zu verbinden, die beide das IP Netz 192.168.6.0/24 benutzen.

transport net das Transport Netzwerk besteht aus zwei Elementen:

- der Verbindung zwischen zwei OpenVPN-Daemons, beschrieben durch *remote_host:remote_port* und *(local_host:)local_port*. Das entspricht den OpenVPN Einstellungen `OPENVPN_x_REMOTE_HOST`, `OPENVPN_x_REMOTE_PORT`, `OPENVPN_x_LOCAL_HOST` und `OPENVPN_x_LOCAL_PORT`.
- und dem Tunnel, über den die Verbindung zwischen den OpenVPN-Daemons etabliert wird, beschrieben durch *local_vpn_ip/remote_vpn_ip*. Dies entspricht dann wieder `OPENVPN_x_LOCAL_VPN_IP` und `OPENVPN_x_REMOTE_VPN_IP`. Die beiden VPN IP-Adressen dürfen sich dabei in keinem anderen, den beiden Routern bekannten Netzen liegen.

input_list, forward_list Pakete, die über den Tunnel gehen sollen, müssen zuerst durch den Paketfilter. Dieser erlaubt standardmäßig nur ICMP-Nachrichten (z. B. ping), die man zum Testen des Tunnels verwenden kann. Alles andere muß erst explizit erlaubt werden, im einfachsten Falle durch

```
OPENVPN_x_PF_INPUT_POLICY='ACCEPT'
OPENVPN_x_PF_FORWARD_POLICY='ACCEPT'
```

Bitte denken Sie daran, dass das komplette „Freigeben“ einer VPN Verbindung sicherheitstechnisch sehr bedenklich ist. Benutzen Sie lieber die tmpl: Syntax des Paketfilters, um nur gezielt die Dienste freizugeben, die Sie auch benötigen.

Mehr Einstellungen sind für einen einfachen VPN Tunnel nicht notwendig. Alle weiteren Einstellmöglichkeiten behandeln erweiterte Funktionen oder sind für spezielle Anwendungsfälle gedacht. Sie sollten mit diesen erweiterten Einstellungen erst dann arbeiten, weil der VPN Tunnel mit den minimalen Einstellungen erfolgreich aufgebaut werden kann.

4.15.2. OpenVPN - Konfiguration

Da OpenVPN ziemlich komplex ist, beginnen wir mit der Erklärung der zwingend notwendigen Angaben für jede VPN Verbindung. Erst wenn der fli4l-Router mit diesen Einstellungen eine Verbindung aufgebaut hat, sollten Sie sich daran wagen die erweiterten Konfigurationsmöglichkeiten von OpenVPN zu nutzen.

OPT_OPENVPN Default: `OPT_OPENVPN='no'`

Mit `'yes'` wird das OpenVPN Paket aktiviert. Die Einstellung `'no'` deaktiviert das OpenVPN Paket komplett.

OPENVPN_x_REMOTE_HOST Default: `OPENVPN_x_REMOTE_HOST="`

Die IP-Adresse oder eine DNS-Adresse der OpenVPN Gegenstelle. Bei einem Roadwarrior (Seite 207) muss diese Zeile komplett fehlen. Wird diese Einstellung weggelassen, wartet OpenVPN auf einen Verbindungsaufbau und versucht nicht selbstständig die Verbindung aufzubauen.

OPENVPN_x_REMOTE_HOSTS_N Default: `OPENVPN_x_REMOTE_HOSTS_N='0'`

Bei der Benutzung von dynamischen DNS Diensten passiert es leider ab und an, dass ein Dienst nicht 100% zuverlässig funktioniert. Daher macht es in diesen Fällen Sinn, einfach zwei oder mehr DynDNS Dienste zu benutzen und seine aktuelle IP-Adresse bei allen Diensten gleichzeitig zu registrieren. Damit OpenVPN in diesem Fall auch alle DynDNS Namen durchgehen kann, muss hier noch die Liste der *zusätzlichen* DNS Namen eingegeben werden. Zusammen mit `OPENVPN_x_REMOTE_HOST` ergibt sich dann die Liste der DynDNS Adressen, die OpenVPN in zufälliger Reihenfolge zu kontaktieren versucht. Der Eintrag `OPENVPN_x_REMOTE_HOST` muss also weiterhin vorhanden sein!

OPENVPN_x_REMOTE_HOST_x Default: `OPENVPN_x_REMOTE_HOST_x="`

Es gilt die gleiche Beschreibung wie unter `OPENVPN_x_REMOTE_HOST` (Seite 187).

OPENVPN_x_REMOTE_PORT Default: `OPENVPN_x_REMOTE_PORT="`

Jede OpenVPN Verbindung braucht eine auf dem fli4l-Router bisher nicht benutzte Portadresse. Es empfiehlt sich, einen Port oberhalb von 10000 zu nehmen, da dort normalerweise keine häufig benutzen Ports liegen. Wenn Sie eine Verbindung für eine Gegenstelle bereitstellen wollen, die eine wechselnde IP-Adresse hat und über keine DynDNS Adresse verfügt, lassen Sie diesen Eintrag genau wie `OPENVPN_x_REMOTE_HOST` komplett weg.

OPENVPN_x_LOCAL_HOST Default: `OPENVPN_x_LOCAL_HOST="`

Gibt an, an welche IP-Adresse OpenVPN gebunden werden soll. Bei Verbindungen über das Internet sollte dieser Eintrag leer bleiben oder komplett weggelassen werden. Wird hier eine IP-Adresse angegeben, horcht OpenVPN nur auf dieser IP-Adresse auf eingehende Verbindungsanfragen. Wenn Sie eine WLAN Verbindung absichern wollen, sollten Sie hier die IP-Adresse der WLAN Karte vom fli4l-Router eintragen.

OPENVPN_x_LOCAL_PORT Default: `OPENVPN_x_LOCAL_PORT="`

Gibt die Portnummer an, auf der der lokale OpenVPN Daemon horcht. Für jede OpenVPN Einstellung benötigen Sie einen dafür reservierten Port, d.h. dieser Port kann nur von dieser einer OpenVPN Verbindung benutzt werden und darf auch von keiner anderen Software auf dem fli4l-Router benutzt werden. Die Einstellungen `OPENVPN_x_REMOTE_PORT` und `OPENVPN_x_LOCAL_PORT` jeder OpenVPN Verbindung müssen zusammenpassen! Wenn Sie auf einer Seite des Tunnel `OPENVPN_x_REMOTE_PORT='10111'` setzen *muss* die andere zwingend auf `OPENVPN_x_LOCAL_PORT='10111'` gesetzt werden.

Nochmal: Es ist sehr wichtig, diese Einstellungen auf die jeweilige OpenVPN Gegenstelle anzupassen, sonst ist eine Verbindung zwischen den OpenVPN Partnern nicht möglich.

Damit OpenVPN auf eingehende Verbindungen horchen kann, öffnet OpenVPN selbstständig die Ports im Paketfilter, die unter `OPENVPN_x_LOCAL_PORT` angegeben werden. Wenn dies nicht gewünscht wird, können Sie dies unter `OPENVPN_DEFAULT_OPEN_OVPNPORT` (Seite 194) anpassen. Es ist *nicht* notwendig den Eintrag `OPENVPN_DEFAULT_OPEN_OVPNPORT='yes'` zu setzen, da das die Standardeinstellung ist!

Es ist nicht möglich OpenVPN auf Ports kleiner als 1025 horchen zu lassen. Wenn Sie z.B. einen OpenVPN als tcp-server auf Port 443 (der https Port) konfigurieren wollen, müssen Sie den Port 443 per Paketfilter an einen Port über 1024 weiterleiten. Wenn Sie

z.B. den OpenVPN auf Port 5555 horchen lassen und den Port 443 weiterleiten wollen, muss folgendes in die PF_PREROUTING eingetragen werden.

```
PF_PREROUTING_5='tmpl:https dynamic REDIRECT:5555'
```

OPENVPN_x_SECRET Default: OPENVPN_x_SECRET="

OpenVPN benötigt zum Verschlüsseln der OpenVPN Verbindung ein sogenanntes Keyfile. Dieses Keyfile kann unter Windows oder Linux direkt mit OpenVPN erzeugt werden. Für Anfänger bietet es sich an, die Windows Software von OpenVPN zu installieren oder die OpenVPN WebGUI zu benutzen. Wenn Sie OpenVPN unter Windows nicht einsetzen wollen, sondern nur die für OpenVPN benötigen Keyfiles erstellen wollen, reicht es, die Punkte *OpenVPN User-Space Components*, *OpenSSL DDLs*, *OpenSSL Utilities*, *Add OpenVPN to PATH* und *Add Shortcuts to OpenVPN* zu installieren. Mit dem Menüpunkt *Generate a static OpenVPN key*, den Sie im Startmenü unter OpenVPN finden, können dann die benötigten Keydateien erzeugt werden. Nach dem Aufruf des Menüpunktes kommt die Meldung „Randomly generated 2048 bit key written to C:/Programme/OpenVPN/config/key.txt“. Die erstellte key.txt Datei ist das benötigte Keyfile. Kopieren Sie diese Datei einfach in das Verzeichnis <config>/etc/openvpn und bennenen Sie die key.txt entsprechend um, so dass der Dateiname aussagekräftig wird. Sie können ein Keyfile auch automatisch vom fli4l-Router erstellen lassen, wenn Sie OPENVPN_CREATE_SECRET auf 'yes' stellen und den fli4l-Router neu starten. Wenn Sie also das erste Mal OpenVPN konfigurieren, tragen Sie alle Daten in die Konfigdatei ein und setzen entweder OPENVPN_DEFAULT_CREATE_SECRET (Seite 194) auf 'yes', wenn Sie gleich für alle OpenVPN Verbindungen neue Keyfiles erzeugen wollen oder nur für die entsprechende OpenVPN Verbindung OPENVPN_x_CREATE_SECRET auf 'yes'. Nach dem Start des fli4l-Routers werden dann die oder das Keyfile(s) automatisch erzeugt und in /etc/openvpn mit dem hier angegebenen Namen abgelegt. Das oder die Keyfile(s) kann dann per scp kopiert oder mit einer Diskette übertragen werden. Sie müssen nach dem Erstellen der Schlüsseldateien die Einstellung wieder auf 'no' setzen, die fli4l-Routerdiskette neu erzeugen und die neu erzeugte Konfiguration starten. Bleibt die Einstellung auf 'yes' werden bei jedem Start des fli4l-Routers neue Schlüsseldateien erzeugt aber kein OpenVPN Daemon gestartet. Es also kann kein Tunnel aufgebaut werden. Sie können OPENVPN_x_CREATE_SECRET auch auf 'webgui' setzen, wenn Sie die WebGUI verwenden möchten um Keyfile(s) zu generieren. Dazu müssen Sie in der WebGUI in die Detailansicht der Verbindung(en) gehen und den Punkt Keymanagement auswählen. Genaueres dazu im Abschnitt 4.15.7

Tipp: Mit dem Kommando

```
fli4l> openvpn --genkey --secret <dateiname>
```

können Sie ein Keyfile auf dem fli4l-Router auch von Hand erstellen.

Die Keyfiles müssen in das Verzeichnis <config>/etc/openvpn kopiert werden, wie in folgendem Bild zu sehen ist. Der Dateiname des Keyfiles ohne den Pfad muss anschließend in OPENVPN_x_SECRET hinterlegt werden. Dann werden die Keyfiles beim Erstellen des opt-Archives mit eingepackt.

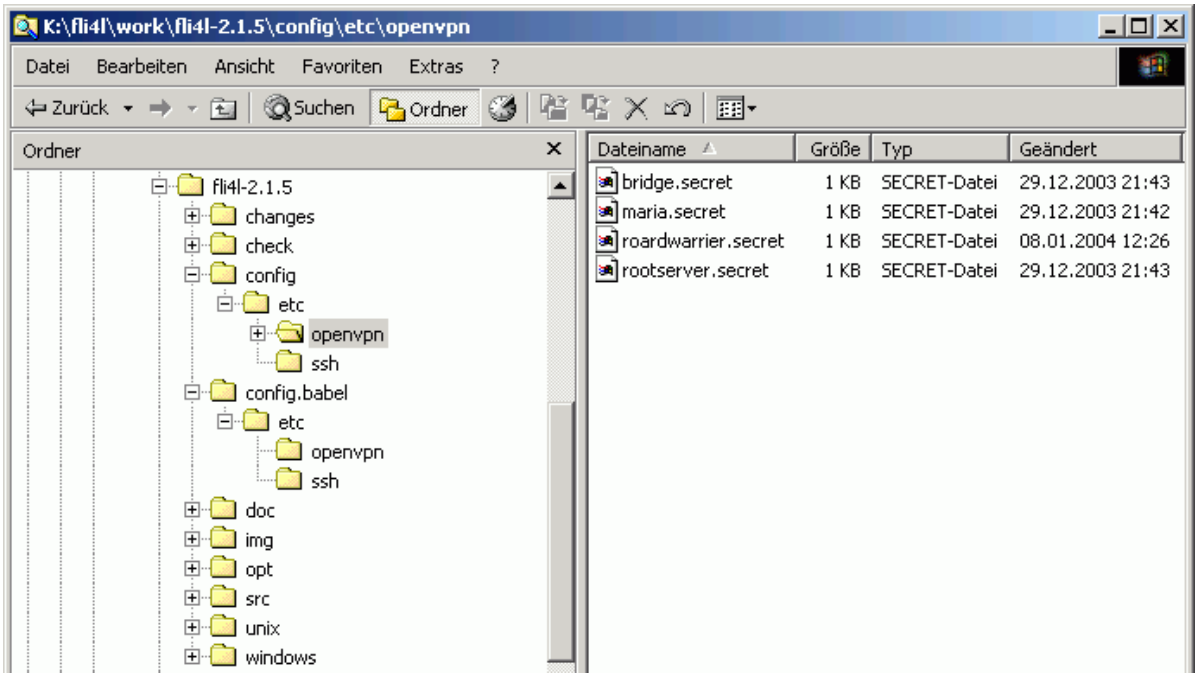


Abbildung 4.2.: fli4l config Directory mit OpenVPN *.secret Dateien

OPENVPN_x_TYPE Default: `OPENVPN_x_TYPE="`

Eine OpenVPN Verbindung kann entweder als Tunnel oder als Bridge benutzt werden. Über einen OpenVPN Tunnel kann ausschliesslich IP-Traffic geroutet werden. Über eine Bridge werden Ethernetframes übertragen, also nicht nur IP-Traffic, sondern z.B. auch IPX oder NetBUI. Wenn OpenVPN als Transport für Ethernetframes benutzt werden soll, wird in jedem Fall noch das advanced_networking Paket benötigt. Bitte bedenken Sie, dass die Benutzung von Bridging über eine DSL Leitung sehr langsam werden kann!

4.15.3. OpenVPN - Bridgekonfiguration

Wenn Sie OpenVPN für eine Bridge benutzen wollen, sind folgende Einträge gültig. Bitte denken Sie daran, dass bei der Benutzung einer Bridge über das Internet der entstehende Broadcasttraffic unter Umständen schon eine relativ hohe Bandbreite benötigt.

Denken Sie daran, dass die folgenden Einstellungen nur gültig sind, wenn der `OPENVPN_x_TYPE` (Seite 190) für diese OpenVPN Verbindung auf `'bridge'` eingestellt wurde! Ausserdem wird eine konfigurierte Bridge aus dem advanced_networking Paket benötigt, an die sich die VPN Verbindung hängen kann.

OPENVPN_x_BRIDGE Default: `OPENVPN_x_BRIDGE="`

Hier wird der Name der Bridge angegeben, an die sich diese OpenVPN Verbindung hängen soll. Wenn also in `BRIDGE_DEV_x_NAME='cuj-br'` steht und sich diese OpenVPN Verbindung an diese Bridge hängen soll, muss hier ebenfalls `'cuj-br'` angegeben werden.

OPENVPN_x_BRIDGE_COST Default: `OPENVPN_x_BRIDGE_COST="`

Wenn Sie STP (siehe http://de.wikipedia.org/wiki/Spanning_Tree oder die Dokumentation im advanced_networking Paket) benutzen, können Sie hier die Kosten der Verbindung angeben.

OPENVPN_x_BRIDGE_PRIORITY Default: `OPENVPN_x_BRIDGE_PRIORITY="`

Wenn Sie STP (siehe http://de.wikipedia.org/wiki/Spanning_Tree oder die Dokumentation im advanced_networking Paket) benutzen, können Sie hier die Priorität der Verbindung angeben.

4.15.4. OpenVPN - Tunnelkonfiguration

OPENVPN_x_REMOTE_VPN_IP Default: `OPENVPN_x_REMOTE_VPN_IP="`

Diese Einstellung ist nur gültig, wenn der OPENVPN_x_TYPE (Seite 190) für diese OpenVPN Verbindung auf `'tunnel'` einstellt wurde!

Die VPN IP-Adresse der OpenVPN Gegenstelle. Die VPN IP-Adressen werden nur zum Routen benötigt und können fast frei gewählt werden. Es gelten dabei folgende Einschränkungen für die Auswahl der VPN IP-Adressen:

- Die IP-Adresse darf an keiner Stelle im lokalen Netz benutzt werden. Sie darf also nicht im Subnetz des fli4l-Routers liegen.
- Die IP-Adresse darf für kein lokales Netzwerkdevice benutzt werden.
- Die IP-Adresse darf nicht zu einem Netzwerk gehören, das mit `IP_ROUTE_x` geroutet wird.
- Die IP-Adresse darf nicht zu einem Netzwerk gehören, das mit `ISDN_CIRC_ROUTE_x` geroutet wird.
- Die IP-Adresse darf nicht zu einem Netzwerk gehören, das mit `CIPE_ROUTE_x` geroutet wird.
- Die IP-Adresse darf nicht zu einem Netzwerk gehören, das mit `OPENVPN_ROUTE_x` geroutet wird.
- Die IP-Adresse darf nicht zu einem Netzwerk gehören, das auf irgendeine andere Weise zum fli4l Netzwerk gehört oder vom fli4l-Router geroutet wird.

Wie Sie sehen darf die VPN IP-Adresse nirgends sonst benutzt werden. Bevor Sie mit der Konfiguration von OpenVPN beginnen, sollten Sie sich ein Netz suchen, was von keinem Netzwerk benutzt wird, in das Sie eine VPN Verbindung aufbauen wollen. Das Netzwerk sollte auch unbedingt zu einem der privaten Netze gehören (siehe http://ftp.univie.ac.at/netinfo/rfc/rfc1597.txt).

OPENVPN_x_LOCAL_VPN_IP Default: `OPENVPN_x_LOCAL_VPN_IP="`

Diese Einstellung ist nur gültig, wenn der OPENVPN_x_TYPE (Seite 190) für diese OpenVPN Verbindung auf `'tunnel'` einstellt wurde.

Die IP-Adresse, die das lokale OpenVPN tunX Device bekommt. Für die Auswahl der IP-Adresse gelten die gleichen Einschränkungen wie bei OPENVPN_x_REMOTE_VPN_IP (Seite 191).

Es ist übrigens möglich, für alle lokalen OpenVPN Verbindungen die gleiche IP-Adresse bei OPENVPN_x_LOCAL_VPN_IP zu benutzen. So ist es problemlos möglich, dass ein Host in einem VPN immer die gleiche IP-Adresse benutzt. Das vereinfacht die Paketfilterregeln unter Umständen dramatisch.

OPENVPN_x_ROUTE_N Default: OPENVPN_x_ROUTE_N="

Diese Einstellung ist nur gültig, wenn der OPENVPN_x_TYPE (Seite 190) für diese OpenVPN Verbindung auf 'tunnel' einstellt wurde.

Die angegebenen Routen werden automatisch von OpenVPN gesetzt, sobald OpenVPN gestartet wird. Es können bis zu 50 Netze über eine OpenVPN Verbindung geroutet werden. Sie müssen aber für jedes zu routende Netzwerk einen OPENVPN_x_ROUTE_x Eintrag erzeugen.

Bitte beachten Sie, dass Sie notwendige Paketfilterregeln in der OPENVPN_PF_FORWARD_x OPENVPN_PF_INPUT_x bzw selber setzen müssen. OpenVPN erlaubt nur ICMP über die VPN Verbindungen und verbietet allen anderen Datenverkehr. Details finden Sie unter OPENVPN_x_PF_INPUT_N (Seite 201) und OPENVPN_x_PF_FORWARD_N (Seite 201).

OPENVPN_x_ROUTE_x Default: OPENVPN_x_ROUTE_x="

Sie müssen hier die Netze angeben, die Sie über die OpenVPN Gegenstelle erreichen wollen. Sind hinter der OpenVPN Gegenstelle z.B. die Netzwerke 192.168.33.0/24 und 172.18.0.0/16 erreichbar und sollen diese über den OpenVPN Tunnel erreicht werden, müssen Sie diese beiden Netze jeweils einzeln unter OPENVPN_x_ROUTE_x eintragen. Es können hier auch Hostrouten (/32) eingetragen werden.

Wenn die Defaultroute über einen OpenVPN Tunnel gesetzt werden soll, geben sie bitte 0.0.0.0/0 und ein optionales Flag als Route an. OpenVPN kennt verschiedene Möglichkeiten die Defaultroute einzurichten, die sie mit dem Flag auswählen können. Jede Methode, die Defaultroute einzurichten, hat ihre Vor- und Nachteile. Im Moment unterstützt OpenVPN folgende Flags:

local Das *local* Flag sollten sie wählen, wenn die OpenVPN Gegenstelle innerhalb eines direkt von ihrem fli4l-Router erreichbaren Subnetz liegt. Das ist z.B. oft bei einer Defaultroute mit OpenVPN über WLAN der Fall.

def1 Mit diesem Flag werden zusätzliche zu einer Hostroute an die OpenVPN Gegenstelle zwei neue Routen, 0.0.0.0/1 und 128.0.0.0/1, eingetragen. Diese Routen übernehmen die Funktion einer Defaultroute und über diese Routen wird dann der komplette (verschlüsselte) Traffic an die OpenVPN Gegenstelle (die noch über die Hostroute erreichbar ist) geroutet.

Lassen Sie das optionale Flag weg wählt OpenVPN die Methode aus, wie die Defaultroute umgesetzt wird. Die Auswahl der Methode erfolgt dabei über die OpenVPN Version, im Moment wird als Standardeinstellung *local* benutzt.

```
OPENVPN_1_ROUTE_N='2'
OPENVPN_1_ROUTE_1='192.168.33.0/24'
OPENVPN_1_ROUTE_2='172.18.0.0/16'
```

4.15.5. OpenVPN - Delegation von DNS und Reverse-DNS

OPENVPN_x_DOMAIN Default: `OPENVPN_x_DOMAIN="`

Über diesen Parameter gibt man die Remote Domain an. Diese Variable kann mehrere Domains enthalten die durch Leerzeichen getrennt angegeben werden müssen. Wenn nur dieser Parameter gesetzt ist (ohne Angabe eines zusätzlichen DNS-Servers) wird davon ausgegangen, daß der DNS Server auf der IP des gegenüberliegenden Tunnelendes lauscht (siehe `OPENVPN_x_REMOTE_VPN_IP` (Seite 191)). Dazu müssen auf dem Remote Router natürlich eingehende DNS-Anfragen erlaubt werden. (z.B. via `OPENVPN_x_INPUT_y='tmpl:dns ACCEPT'`)

OPENVPN_x_DOMAIN_x Default: `OPENVPN_x_DOMAIN_x="`

Den verschiedenen Subnetzen können auch verschiedene Domains zugeordnet sein. Hier kann man pro `OPENVPN_x_ROUTE_y` eine entsprechende Domain konfigurieren. Sollte ein dazugehöriges `OPENVPN_x_DNSIP_y` existieren, so wird dieser Server benutzt, andernfalls der unter `OPENVPN_x_DNSIP` angegebene. Die Wirkung ist dann die selbe wie mit `OPENVPN_x_DOMAIN`, allerdings eignet sich diese Methode auch zur Dokumentation.

OPENVPN_x_DNSIP Default: `OPENVPN_x_DNSIP="`

Ist der Tunnelendpunkt nicht der zuständige DNS-Server, so kann hier die IP des zuständigen DNS-Servers angegeben werden. Ist nichts angegeben, wird die unter `OPENVPN_x_REMOTE_VPN_IP` (Seite 191) angegebene IP benutzt.

OPENVPN_x_DNSIP_x Default: `OPENVPN_x_DNSIP_x="`

Verschiedene geroutete Subnetze können auch durch verschiedene DNS-Server abgedeckt sein - hier kann man pro `OPENVPN_x_ROUTE_x` (Seite 192) einen eigenen zuständigen Server definieren.

4.15.6. Experteneinstellungen

Die in diesem Kapitel beschriebenen Einstellungen sind alle optional und sollten nur verändert werden, wenn die OpenVPN Verbindung mit den Standardeinstellungen funktioniert und Optimierungen (beispielsweie ein anderer Verschlüsselungsalgorithmus) vorgenommen werden sollen.

Alle nachfolgend beschriebenen `OPENVPN_DEFAULT_` Einstellungen sind optional, d.h. diese Optionen brauchen nicht in die openvpn.txt Konfigurationsdatei geschrieben werden. Fehlt der entsprechende Eintrag in der openvpn.txt Datei, wird vom OpenVPN Startskript der hier angegebene Defaultwert benutzt. Wenn Sie nicht vorhaben die Standardwerte der Vorgaben zu änden, schreiben Sie diese auch nicht in die openvpn.txt Konfigurationsdatei!

OPENVPN_DEFAULT_CIPHER Default: `OPENVPN_DEFAULT_CIPHER='BF-CBC'`

Eine der verfügbaren Verschlüsselungsmethoden. Die Verschlüsselungsmethode `'BF-CBC'` wird von allen OpenVPN Versionen (auch nicht fli4l spezifischen Versionen) als Standardeinstellung benutzt.

OPENVPN_DEFAULT_COMPRESS Default: `OPENVPN_DEFAULT_COMPRESS='yes'`

OpenVPN benutzt eine adaptive LZO Datenkomprimierung, um die Bandbreite einer Verbindung zu erhöhen. Adaptiv bedeutet, dass OpenVPN selbstständig erkennt, wenn

z.B. bereits gepackte ZIP Dateien über eine OpenVPN Verbindung geschickt werden. In einem solchen Fall wird die Datenkomprimierung solange abgeschaltet, bis wieder Daten übertragen werden, die auch von einer Datenkomprimierung profitieren. Es gibt fast nie einen Grund die Datenkomprimierung zu deaktiveren, da dadurch die Bandbreite quasi kostenlos erhöht wird. Einziger Nachteil der Datenkomprimierung ist eine geringe Erhöhung der Latenzzeit von wenigen Millisekunden. Für Online-Games via VPN, bei denen die Reaktionszeit (”guter” ping) entscheidend ist, wäre es also unter Umständen sinnvoll die Datenkomprimierung abzuschalten.

OPENVPN_DEFAULT_CREATE_SECRET Default: `OPENVPN_DEFAULT_CREATE_SECRET='no'`

Mit dieser Einstellung erstellt OpenVPN automatisch Keyfiles beim Start des fli4l-Routers. Die entsprechende OpenVPN Verbindung wird allerdings nicht gestartet. Diese Einstellung ist nützlich, um automatisch die notwendigen Keyfiles für OpenVPN zu erstellen. Für Details lesen Sie bitte den Punkt `OPENVPN_x_SECRET` (Seite 189) nach.

OPENVPN_DEFAULT_DIGEST Default: `OPENVPN_DEFAULT_DIGEST='SHA1'`

Hier wird eine der verfügbaren Prüfsummen eingetragen. OpenVPN nutzt als Standardeinstellung die Prüfsummenmethode `'SHA1'`.

OPENVPN_DEFAULT_FLOAT Default: `OPENVPN_DEFAULT_FLOAT='yes'`

Bei OpenVPN Verbindungen mit Gegenstellen, die eine DynDNS Adresse benutzen, ist es jederzeit möglich, dass sich die IP-Adresse der OpenVPN Gegenstelle ändert. Damit OpenVPN auch die geänderte IP-Adresse akzeptiert, muss `OPENVPN_DEFAULT_FLOAT` auf `'yes'` eingestellt werden. Mit der Einstellung `'no'` wird die Änderung der IP-Adresse nicht erlaubt. Das ist in der Regel nur bei WLAN Verbindungen oder OpenVPN Verbindungen mit Gegenstellen, die eine statische IP-Adresse (wie z.B. die rootserver diverser Anbieter) haben, sinnvoll. Sie können diese Einstellung auch wie alle anderen `OPENVPN_DEFAULT_` Einstellungen für eine bestimmte OpenVPN Verbindung überschreiben.

OPENVPN_DEFAULT_KEYSIZE Default: `OPENVPN_DEFAULT_KEYSIZE="`

Die Schlüssellänge (=KEYSIZE) hängt von der verwendeten Verschlüsselungsmethode ab. Ändern Sie diese Einstellung nur, wenn Sie mit einer OpenVPN Gegenstelle zusammenarbeiten müssen, die nicht den Standardwert benutzt oder deren Einstellung Sie nicht beeinflussen können. Können Sie die Schlüssellänge selbst bestimmen, sollte dieser Wert immer leer bleiben. OpenVPN wendet dann eine optimale Schlüssellänge für die jeweilige Verschlüsselungmethode an.

OPENVPN_DEFAULT_OPEN_OVPNPORT Default: `OPENVPN_DEFAULT_OPEN_OVPNPORT='yes'`

Damit eine OpenVPN Gegenstelle mit Ihnen Kontakt aufnehmen kann, müssen Sie den Paketfilter Ihres fli4l-Routers entsprechend anpassen. In der Regel müssen Sie also auf allen TCP oder UDP Ports, je nach `OPENVPN_x_PROTOCOL` Einstellung, auf denen ein OpenVPN horcht, die `PF_INPUT_x` (Seite 41) in der base.txt anpassen. Mit der Einstellung `'yes'` werden diese Paketfilterregeln automatisch generiert. Bei einzelnen Verbindungen kann es aber durchaus Sinn machen, diese Einstellung auf `'no'` zu setzen und selber entsprechende Paketfilterregeln zu definieren.

OPENVPN_DEFAULT_ALLOW_ICMPPING Default: `OPENVPN_DEFAULT_ALLOW_ICMPPING='yes'`

Mit `'yes'` wird der Paketfilter für die entsprechende Verbindung so konfiguriert, dass ping Datenpakete den Filter passieren dürfen. Wenn es keinen sehr wichtigen Grund gibt, sollte der ICMP Ping immer zugelassen werden. Diese Einstellung hat *nichts* mit der Pingoption von OpenVPN zu tun!

OPENVPN_DEFAULT_PF_DMZ_TYPE Default: `OPENVPN_DEFAULT_PF_DMZ_TYPE='none'`

Hier kann bei Verwendung einer DMZ festgelegt werden, ob die OpenVPN-Verbindungen zu den grünen Netzwerken gehören, oder nicht. Zulässig sind die Werte 'green' und 'none'.

OPENVPN_DEFAULT_PF_INPUT_LOG Default: `OPENVPN_DEFAULT_PF_INPUT_LOG='BASE'`

Mit `'yes'` oder `'no'` wird eingestellt, ob der Paketfilter eingehende Datenpakete in der INPUT Liste, die für die entsprechende VPN Verbindung gedacht waren mitschreiben soll, wenn das Datenpaket abgelehnt wird. Mit der Einstellung `'BASE'` wird die Einstellung von `'PF_INPUT_LOG='` aus der base.txt übernommen.

OPENVPN_DEFAULT_PF_INPUT_POLICY Default: `OPENVPN_DEFAULT_PF_INPUT_POLICY='REJECT'`

Diese Einstellung entspricht `'PF_INPUT_POLICY='` (Seite 57) aus der base.txt. Zusätzlich zu den dort möglichen Einstellungen, kann mit `'BASE'` die Einstellung von `'PF_INPUT_POLICY='` aus der base.txt übernommen werden.

OPENVPN_DEFAULT_PF_FORWARD_LOG Default: `OPENVPN_DEFAULT_PF_FORWARD_LOG='BASE'`

Mit `'yes'` oder `'no'` wird eingestellt, ob der Paketfilter eingehende Datenpakete in der FORWARD Liste, die für die entsprechende VPN Verbindung gedacht waren mitschreiben soll, wenn das Datenpaket abgelehnt wird. Mit der Einstellung `'BASE'` wird die Einstellung von `'PF_FORWARD_LOG='` aus der base.txt übernommen.

OPENVPN_DEFAULT_PF_FORWARD_POLICY Default: `OPENVPN_DEFAULT_PF_FORWARD_POLICY='REJECT'`

Diese Einstellung entspricht `'PF_FORWARD_POLICY='` (Seite 59) aus der base.txt. Zusätzlich zu den dort möglichen Einstellungen kann mit `'BASE'` die Einstellung von `'PF_FORWARD_POLICY='` aus der base.txt übernommen werden.

OPENVPN_DEFAULT_PING Default: `OPENVPN_DEFAULT_PING='60'`

Um einen einmal aufgebauten Tunnel offen zu halten und zu erkennen, ob die OpenVPN Gegenstelle noch erreichbar ist, wird in dem angegebenen Intervall in Sekunden ein kryptografisch gesicherter ping über die Leitung geschickt. Mit der Einstellung `'off'` schickt OpenVPN kein ping über die Leitung. Mit der Einstellung `'off'` werden nur dann Daten über den VPN Tunnel geschickt wenn auch tatsächlich Nutzdaten über die VPN Tunnel übertragen werden.

OPENVPN_DEFAULT_PING_RESTART Default: `OPENVPN_DEFAULT_PING_RESTART='180'`

Wird in dem angegebenen Intervall in Sekunden kein OpenVPN ping erfolgreich über die VPN Verbindung geschickt oder werden keine anderen Daten übertragen, wird die entsprechende OpenVPN Verbindung neu gestartet. Der Wert von `OPENVPN_DEFAULT_PING_RESTART` muss immer größer sein als der Wert von `OPENVPN_DEFAULT_PING`. Die Einstellung `'off'` unterbindet den automatischen Neustart einer OpenVPN Verbindung.

OPENVPN_DEFAULT_RESOLV_RETRY Default: `OPENVPN_DEFAULT_RESOLV_RETRY='infinite'`

Sind in `OPENVPN_x_REMOTE_HOST` oder `OPENVPN_x_LOCAL_HOST` DNS Namen statt IP-Adressen hinterlegt, dann müssen die Namen beim Start einer OpenVPN Verbindung erst zu IP-Adressen aufgelöst werden. Sollte dies fehlschlagen, versucht OpenVPN für die angegebene Zeit in Sekunden die DNS Adresse erneut aufzulösen. Gelingt dies schlußendlich nicht in der vorgegebenen Zeit, kommt keine OpenVPN Verbindung zustande. Mit der Angabe `'infinite'` (= unendlich) wird unendlich lange versucht, die DNS Auflösung vorzunehmen. Diese Einstellung sollte bis auf besondere Fälle nicht verändert werden!

OPENVPN_DEFAULT_RESTART Default: `OPENVPN_DEFAULT_RESTART='ip-up'`

Nach einer Trennung der Verbindung ist es sinnvoll, dass der entsprechende OpenVPN Tunnel sofort neu gestartet wird, damit die Unterbrechung des Tunnels möglichst kurz ist. Für alle OpenVPN Verbindungen, die über eine Wählverbindung wie DSL oder ISDN laufen, sollte hier `'ip-up'` eingetragen werden. OpenVPN Verbindungen über eine WLAN Strecke dagegen sollten hier `'never'` eintragen. In diesem Fall wird die Verbindung nicht nach einer Neueinwahl wieder gestartet, da ja die WLAN Verbindung unabhängig von einer Einwahl ist. Wenn ein OpenVPN Tunnel über eine ISDN Wählverbindung mit `ISDN_CIRC_x_TYPE='raw'` aufgebaut werden soll, muss hier `'raw-up'` eingetragen werden.

OPENVPN_DEFAULT_PROTOCOL Default: `OPENVPN_DEFAULT_PROTOCOL='udp'`

Mit dieser Varablen wird festgelegt, welche Protokollvariante als Default benutzt werden soll. Normalerweise ist UDP eine sehr gute Wahl, allerdings gibt es manchmal nur die Möglichkeit mit TCP zu arbeiten. Der dadurch entstehende Overhead ist allerdings beachtlich. Mögliche Einstellungen sind `'udp'`, `'tcp-server'` oder `'tcp-client'`. Die `'tcp-server'` oder `'tcp-client'` Einstellungen sind in der Regel nur dann sinnvoll, wenn ein VPN Tunnel durch div. andere Paketfilter oder andere Tunnel aufgebaut werden soll. Wenn Sie keine Spezialfälle behandeln müssen, sollten Sie *immer* den Standardwert `'udp'` benutzen.

OPENVPN_DEFAULT_START Default: `OPENVPN_DEFAULT_START='always'`

Eine OpenVPN Verbindung kann entweder immer (`='always'`) oder später von Hand (`='on-demand'`) gestartet werden. Sie können also bestimmte OpenVPN Verbindungen erst bei Bedarf über die OpenVPN WebGUI (siehe 4.15.7) starten. Alternativ ist der Start aber auch jederzeit über die fli4l Konsole möglich. Melden Sie sich dazu auf der fli4l Konsole an und führen Sie folgende Befehle direkt auf der Konsole aus:

```
cd /etc/openvpn
openvpn --config name.conf --daemon openvpn-name
```

Damit wird ein OpenVPN Tunnel gestartet, der ab sofort im Hintergrund läuft. Anstelle `name.conf` nehmen Sie natürlich den Namen Ihrer Konfigurationsdatei in dem `/etc/openvpn` Verzeichnis.

OPENVPN_DEFAULT_VERBOSE Default: `OPENVPN_DEFAULT_VERBOSE='2'`

Diese Variable gibt an, wie gesprächig OpenVPN sein soll. Wenn eine VPN Verbindung einwandfrei läuft, ist es möglich diesen Wert auf `'0'` zu setzen, um alle Meldungen zu unterbinden. Für die ersten Tests ist ein Wert von `'3'` sinnvoll. Noch größere Werte erhöhen

die Debugmeldungen und helfen unter Umständen Fehler zu finden. Der Maximalwert beträgt `'11'`.

OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE Default:
`OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE='100'`

Diese Variable gibt an, wie viele Log Zeilen gespeichert werden sollen. Dieses Log kann dann über die WebGUI (Seite 202) abgefragt werden.

OPENVPN_DEFAULT_MUTE_REPLAY_WARNINGS Default:
`OPENVPN_DEFAULT_MUTE_REPLAY_WARNINGS='no'`

Mit dieser Variablen wird eingestellt, ob beim Empfang doppelter Pakete eine Warnung im Log ausgegeben werden soll, da dies Hinweise auf ein Sicherheitsproblem im Netzwerk sein können. Insbesondere bei schwachen WLAN-Verbindungen, kann es aber häufig passieren, dass Pakete doppelt gesendet werden. Dann ist es sinnvoll die Warnungen auszustellen, damit diese nicht das Log füllen. Die Einstellung dieser Variablen hat *keinen* Einfluss auf die Sicherheit der OpenVPN Verbindung.

OPENVPN_DEFAULT_MSSFIX Default: `OPENVPN_DEFAULT_MSSFIX="`

Mit der MSSFIX Einstellung wird die Größe der TCP Datenpakete über die VPN Verbindung vorgegeben. Mit `OPENVPN_DEFAULT_MSSFIX='0'` wird diese Option ausgeschaltet. Wird eine Fragmentgröße angegeben und der MSSFIX Eintrag leer gelassen, wird automatisch die Größe die Fragmentgröße benutzt. Diese Einstellung funktioniert nur, wenn `OPENVPN_x_PROTOCOL='udp'` gesetzt wird.

OPENVPN_DEFAULT_FRAGMENT Default: `OPENVPN_DEFAULT_FRAGMENT='1300'`

Aktiviert die interne Fragmentierung von OpenVPN mit einer Paketgröße von x Bytes. Diese Einstellung funktioniert nur, wenn `OPENVPN_x_PROTOCOL='udp'` gesetzt wird.

Mit `OPENVPN_DEFAULT_FRAGMENT='0'` wird die Fragmentierung komplett deaktiviert.

OPENVPN_DEFAULT_TUN_MTU Default: `OPENVPN_DEFAULT_TUN_MTU='1500'`

Stellt die MTU des virtuellen OpenVPN Adapters auf x Bytes ein. Diese Option sollte nur geändert werden, wenn man weiss was man macht. Es ist meistens sinnvoller, erst mit den Fragment oder MSSFIX Optionen zu arbeiten.

OPENVPN_DEFAULT_TUN_MTU_EXTRA Default: `OPENVPN_DEFAULT_TUN_MTU_EXTRA="`

Wenn bei `OPENVPN_x_PROTOCOL='bridge'` eingestellt wird, werden 32 Bytes als extra Speicher für die Verwaltung der Puffers für das tap Gerät reserviert. Bei `OPENVPN_x_PROTOCOL='tunnel'` wird kein extra Speicher reserviert. Diese Einstellung wirkt sich nur auf den Speicherbedarf im Router aus und hat keinen Einfluss auf die Datenmenge die über den Tunnel verschickt wird.

OPENVPN_DEFAULT_LINK_MTU Default: `OPENVPN_DEFAULT_LINK_MTU="`

Stellt die MTU der OpenVPN Verbindung auf x Bytes ein. Diese Option sollte nur benutzt werden, wenn man weiss was man macht. Es ist meistens sinnvoller erst mit den Fragment oder MSSFIX Optionen zu arbeiten.

OPENVPN_DEFAULT_SHAPER Default: `OPENVPN_DEFAULT_SHAPER="`

Begrenzt die *ausgehende* Bandbreite des Tunnel auf die angegebene Anzahl von Bytes pro Sekunde. Möglich sind Werte von 100 Bytes bis zu 100000000 Bytes. Bei Werten bis 1000 Bytes pro Sekunde sollten Sie die MTU der Verbindung reduzieren, sonst steigen die ping Zeiten stark an. Wenn Sie einen Tunnel auf eine Bandbreite in beide Richtungen begrenzen wollen, müssen Sie dies auf beiden Seiten getrennt einstellen.

In der aktuellen OpenVPN Version funktioniert das Shaping nicht korrekt, d.h. die Übertragungsrate durch einen mittels Shapping konfigurierten Tunnel ist unter Umständen extrem schwankend bzw. der Durchsatz bricht total ein. Das Problem kann je nach eingesetzer Hardware auftreten zu komplett unterschiedlichen Verhalten führen. Im Moment sollte die Shappingfunktion mit Vorsicht behandelt werden und im Zweifel bei jeder Änderung ausgiebig getestet werden.

OPENVPN_EXPERT Default: `OPENVPN_EXPERT='no'`

Der Expert-Mode erlaubt Ihnen, native Openvpn Konfigurationsdateien zu nutzen. Diese müssen im Konfigurationsordner unter etc/openvpn abgelegt werden. Alle dort liegenden Dateien werden auf den Router übertragen.

Der Expert-Mode ignoriert die restlichen Konfigurationseinstellungen. Deshalb muss `OPENVPN_N='0'` eingestellt werden.

Der Expert-Mode richtet keinerlei Firewall-Regeln ein. Diese müssen Sie manuell in die base.txt eintragen.

Die folgenden OpenVPN Optionen gelten nur für die jeweilige OpenVPN Verbindung. Auch hier gibt es nur wenige Angaben, die zwingend sind. Die meisten Optionen können einfach weggelassen werden. Für alle Defaultwerte gilt, dass diese von der jeweils gleichlautenden `OPENVPN_DEFAULT_x` Einstellung übernommen werden. Wenn Sie also den entsprechenden `OPENVPN_DEFAULT_` Wert ändern, gilt dieser Defaultwert für alle OpenVPN Verbindungen, die den allgemeinen Defaultwert nicht überschreiben.

OPENVPN_N Default: `OPENVPN_N='0'`

Wieviele OpenVPN Konfigurationen sind in der Konfigurationsdatei aktiv?

OPENVPN_x_NAME Default: `OPENVPN_x_NAME="`

Legt einen bis zu 16 Zeichen langen Namen für die jeweilige OpenVPN Verbindung fest. Unter diesem Namen wird die Konfigurationsdatei im Verzeichnis /etc/openvpn (mit dem Anhang .conf) angelegt. Außerdem erscheint der Name im syslog. Wenn Sie beispielsweise den Namen `'peter'` eintragen, taucht später im syslog der Eintrag 'openvpn-peter' auf. Damit können Sie die verschiedenen OpenVPN Verbindungen gut auseinanderhalten. Der Name darf Buchstaben, Zahlen und das '-' Zeichen enthalten.

OPENVPN_x_ACTIV Default: `OPENVPN_x_ACTIV='yes'`

Wenn man eine OpenVPN Verbindung deaktivieren, aber die Konfiguration nicht löschen will, kann diese OpenVPN Verbindung mit der Einstellung 'no' deaktiviert werden. Die Konfigurationsdaten werden dann zwar in der rc.cfg aufgenommen, aber es wird keine entsprechende OpenVPN Verbindung erzeugt.

OPENVPN_x_CHECK_CONFIG Default: `OPENVPN_x_CHECK_CONFIG='yes'`

Die erweiterten Prüfungen von OpenVPN sind in seltenen Fällen zu streng. Wenn beispielsweise eine ISDN Backupverbindung die gleichen Routingeinträge benutzt wie eine Verbindung, die über das Internet läuft, wird die erweiterte Prüfung diese Verbindungen mit einer Fehlermeldung bedenken. In diesem Fall sollte bei der Backupverbindung die erweitere Prüfung deaktiviert werden. Dazu setzen Sie `OPENVPN_x_CHECK_CONFIG='no'`, um die Prüfungen für diese Verbindung auszuschalten.

OPENVPN_x_CIPHER Default siehe: `OPENVPN_DEFAULT_CIPHER`

Siehe `OPENVPN_DEFAULT_CIPHER` (Seite 193). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_COMPRESS Default siehe: `OPENVPN_DEFAULT_COMPRESS`

Siehe `OPENVPN_DEFAULT_COMPRESS` (Seite 193). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_CREATE_SECRET Default siehe: `OPENVPN_DEFAULT_CREATE_SECRET='no'`

Siehe `OPENVPN_x_SECRET` (Seite 189). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_DIGEST Default siehe: `OPENVPN_DEFAULT_DIGEST`

Siehe `OPENVPN_DEFAULT_DIGEST` (Seite 194). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_FLOAT Default siehe: `OPENVPN_DEFAULT_FLOAT`

Siehe `OPENVPN_DEFAULT_FLOAT` (Seite 194). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_KEYSIZE Default siehe: `OPENVPN_DEFAULT_KEYSIZE`

Siehe `OPENVPN_DEFAULT_KEYSIZE` (Seite 194). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_ISDN_CIRC_NAME Default `OPENVPN_x_ISDN_CIRC_NAME=”`

Gibt an über welchen ISDN Circuit diese OpenVPN Verbindung aufgebaut wird. Hier wird der Name des entsprechenden ISDN Circuits eingetragen, der mit `ISDN_CIRC_x_NAME="` (Seite 151) definiert wird. Der entsprechende ISDN Circuit muss dabei vom Typ `'raw'` sein.

OPENVPN_x_PING Default siehe: `OPENVPN_DEFAULT_PING`

Siehe `OPENVPN_DEFAULT_PING` (Seite 195). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PROTOCOL Default: `OPENVPN_x_PROTOCOL='udp'`

Gibt an mit welchem Protokoll der OpenVPN Tunnel aufgebaut werden soll. Mögliche Einstellungen sind `'udp'`, `'tcp-server'` oder `'tcp-client'`. Die `'tcp-server'` oder `'tcp-client'` Einstellungen sind in der Regel nur dann sinnvoll, wenn ein VPN Tunnel durch diverse andere Paketfilter oder andere Tunnel aufgebaut werden soll. Wenn Sie keine Spezialfälle behandeln müssen, sollten Sie *immer* den Standardwert `'udp'` benutzen.

OPENVPN_x_RESOLV_RETRY Default siehe: `OPENVPN_DEFAULT_RESOLV_RETRY`

Siehe `OPENVPN_DEFAULT_RESOLV_RETRY` (Seite 196). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PING_RESTART Default siehe: `OPENVPN_DEFAULT_PING_RESTART`

Siehe `OPENVPN_DEFAULT_PING_RESTART` (Seite 195). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_START Default siehe: `OPENVPN_DEFAULT_START`

Siehe `OPENVPN_DEFAULT_START` (Seite 196). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_VERBOSE Default siehe: `OPENVPN_DEFAULT_VERBOSE`

Siehe `OPENVPN_DEFAULT_VERBOSE` (Seite 196). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_MANAGEMENT_LOG_CACHE Default siehe: `OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE`

Siehe `OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE` (Seite 197). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_MUTE_REPLAY_WARNINGS Default siehe: `OPENVPN_DEFAULT_MUTE_REPLAY_WARNINGS`

Siehe `OPENVPN_DEFAULT_MUTE_REPLAY_WARNINGS` (Seite 197). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_RESTART Default siehe: `OPENVPN_DEFAULT_RESTART`

Siehe `OPENVPN_DEFAULT_RESTART` (Seite 196). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_ALLOW_ICMPPING Default siehe: `OPENVPN_DEFAULT_ALLOW_ICMPPING`

Siehe `OPENVPN_DEFAULT_ALLOW_ICMPPING` (Seite 195). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_OPEN_OVPNPORT Default siehe: `OPENVPN_DEFAULT_OPEN_OVPNPORT`

Siehe `OPENVPN_DEFAULT_OPEN_OVPNPORT` (Seite 194). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PF_DMZ_TYPE Default siehe: `OPENVPN_DEFAULT_PF_DMZ_TYPE`

Siehe `OPENVPN_DEFAULT_PF_DMZ_TYPE` (Seite 195). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PF_INPUT_LOG Default siehe: `OPENVPN_DEFAULT_PF_INPUT_LOG`

Siehe `OPENVPN_DEFAULT_PF_INPUT_LOG` (Seite 195). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PF_INPUT_POLICY Default siehe: OPENVPN_DEFAULT_PF_INPUT_POLICY

Siehe OPENVPN_DEFAULT_PF_INPUT_POLICY (Seite 195). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PF_INPUT_N Default: OPENVPN_x_PF_INPUT_N='0'

Gibt die Anzahl der folgenden OPENVPN_x_PF_INPUT_x= Einträge an.

OPENVPN_x_PF_INPUT_x Default: OPENVPN_x_PF_INPUT_x="

Wie im base Paket stehen hier die Anweisungen für den Paketfilter. Es wird genau die gleiche Syntax wie in base.txt benutzt. Auch tmpl: und Hostaliase sind möglich. Zusätzlich gibt es noch die Möglichkeit, einige spezielle symbolische Namen zu benutzen. Es werden folgende symbolische Namen unterstützt:

VPNDEV Entspricht dem aktuellen tun Device der jeweiligen OpenVPN Verbindung.

LOCAL-VPN-IP Setzt die IP-Adresse von OPENVPN_x_LOCAL_VPN_IP ein.

REMOTE-VPN-IP Setzt die IP-Adresse von OPENVPN_x_REMOTE_VPN_IP ein.

REMOTE-NET Setzt die IP-Adresse von OPENVPN_x_REMOTE_VPN_IP ein und zusätzlich noch alle Netze die mit OPENVPN_x_ROUTE_x angegeben wurden.

OPENVPN_x_PF_FORWARD_LOG Default siehe: OPENVPN_DEFAULT_PF_FORWARD_LOG

Siehe OPENVPN_DEFAULT_PF_FORWARD_LOG (Seite 195). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PF_FORWARD_POLICY Default siehe: OPENVPN_DEFAULT_PF_FORWARD_POLICY

Siehe OPENVPN_DEFAULT_PF_FORWARD_POLICY (Seite 195). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_PF_FORWARD_N Default: OPENVPN_x_PF_FORWARD_N='0'

Gibt die Anzahl der folgenden OPENVPN_x_PF_FORWARD_x= Einträge an.

OPENVPN_x_PF_FORWARD_x Default: OPENVPN_x_PF_FORWARD_x="

Siehe OPENVPN_x_PF_INPUT_x (Seite 201).

OPENVPN_x_PF_PREROUTING_N Default: OPENVPN_x_PF_PREROUTING_N='0'

Gibt die Anzahl der folgenden OPENVPN_x_PF_PREROUTING_x= Einträge an.

OPENVPN_x_PF_PREROUTING_x Default: OPENVPN_x_PF_PREROUTING_x="

Siehe OPENVPN_x_PF_INPUT_x (Seite 201).

OPENVPN_x_PF_POSTROUTING_N Default: OPENVPN_x_PF_POSTROUTING_N='0'

Gibt die Anzahl der folgenden OPENVPN_x_PF_POSTROUTING_x= Einträge an.

OPENVPN_x_PF_POSTROUTING_x Default: OPENVPN_x_PF_POSTROUTING_x="

Ab fli4l Revision 3.5.0 (oder 3.5.0-rev18133 für tarball Benutzer) ergibt sich eine Änderung im Verhalten. War es früher möglich Einträge in der Form von

```
OPENVPN_1_PF_POSTROUTING_1='MASQUERADE'
```

anzugeben wird ab jetzt die Angabe einer Quell und Zieladresse erzwungen. Dies ist notwendig geworden, weil die POSTROUTING Regeln sonst nicht im vollem Umfang genutzt werden konnten. In den meisten Fällen reicht es einfach die Regeln und IP_NET_x (Seite 39) und REMOTE-NET zu ergänzen.

Siehe OPENVPN_x_PF_INPUT_x (Seite 201).

OPENVPN_x_MSSFIX Default siehe: OPENVPN_DEFAULT_MSSFIX

Siehe OPENVPN_DEFAULT_MSSFIX (Seite 197). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_FRAGMENT Default siehe: OPENVPN_DEFAULT_FRAGMENT

Siehe OPENVPN_DEFAULT_FRAGMENT (Seite 197). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_TUN_MTU Default siehe: OPENVPN_DEFAULT_TUN_MTU

Siehe OPENVPN_DEFAULT_TUN_MTU (Seite 197). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_TUN_MTU_EXTRA Default siehe: OPENVPN_DEFAULT_TUN_MTU_EXTRA

Siehe OPENVPN_DEFAULT_TUN_MTU_EXTRA (Seite 197). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_LINK_MTU Default siehe: OPENVPN_DEFAULT_LINK_MTU

Siehe OPENVPN_DEFAULT_LINK_MTU (Seite 197). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

OPENVPN_x_SHAPER Default siehe: OPENVPN_DEFAULT_SHAPER="

Siehe OPENVPN_DEFAULT_SHAPER (Seite 198). Im Gegensatz zu der Default Einstellung wirkt diese Einstellung nur auf diese OpenVPN Verbindung.

4.15.7. OpenVPN - WebGUI

Seit der Version 2.1.10 ist es möglich über eine WebGUI, die konfigurierten OpenVPN Verbindungen zu starten, stoppen und andere grundlegende Funktionen auszuführen. Dazu wird das mini_httpd Paket benötigt. Zudem muss die Variable OPENVPN_WEBGUI in der openvpn.txt auf 'yes' gesetzt werden. Dann wird der fli4l Weboberfläche der Menüpunkt OpenVPN hinzugefügt. Wählt man diesen Menüpunkt an, erscheint eine Übersicht über die konfigurierten OpenVPN Verbindungen, deren Status und die jeweils zur Verfügung stehenden Aktionen (siehe Abbildung 4.3).

OpenVPN - WebGUI - Verbindungsübersicht

Status: Der Status einer Verbindung wird mit Ampelmännchen symbolisiert. Ein rotes Männchen bedeutet, dass der OpenVPN-Prozess nicht läuft, ein gelbes, dass der Prozess zwar läuft aber (noch) keine Verbindung zur Gegenstelle aufgebaut werden konnte und ein grünes, dass die Verbindung mit der Gegenstelle „steht“. Genauere Informationen über den Status erhält man als Tooltip über dem Ampelmännchen. Das kann insbesondere bei „gelbem“ Status aufschlussreich sein.

Abbildung 4.3.: Verbindungsübersicht

Name: In dieser Spalte steht der Name der OpenVPN-Verbindung wie er in der Konfiguration angegeben wurde. Ein Klick auf den Namen führt in eine Übersicht, in der genauere Informationen zu dieser Verbindung angezeigt werden. Dazu später mehr.

Aktion: Hier sind die zur Verfügung stehenden Aktionen als Buttons symbolisiert. Was sie jeweils bedeuten, erfährt man über Tooltips. Folgende Buttons gibt es:

OpenVPN - WebGUI - Detailansicht einer Verbindung

Statistik: Hier werden ein paar interessante Statistiken angezeigt. Die Statistik kann nur angezeigt werden, wenn die Verbindung gestartet und nicht angehalten ist.

Log: Zeigt die letzten 20 Zeilen des Verbindungslogs. Wenn Sie mehr Zeilen sehen möchten, können Sie auch deren Anzahl angeben und auf 'Anzeigen' klicken. Wenn Sie als Anzahl 'all' einngeben, wird das gesamte Log angezeigt. Dieser Reiter wird nur angezeigt, wenn die Verbindung gestartet ist.

Debug-Log: Zeigt die Ausgabe eines Startvorgangs. Die OpenVPN-Verbindung wird gestartet und die Ausgaben dabei angezeigt. Das ist dann nützlich, wenn die Verbindugn über den Start Button nicht starten will und man dadurch kein normales Log zu sehen bekommt. Dieser Reiter wird nur angezeigt, wenn die Verbindung nicht gestartet ist.

Symbol	Erläuterung
	Der OpenVPN-Prozess wird gestartet und es wird versucht eine Verbindung aufzubauen.
	Der OpenVPN-Prozess wird beendet.
	Die Verbindung wird zurückgesetzt.
	Die Verbindung wird zurückgesetz und auf 'hold' geschaltet. Dann gehen keine Daten mehr über die Verbindung.
	Die Verbindung wird wieder freigegeben. Daten können wieder über die Verbindung fließen.

Tabelle 4.10.: Aktionen der OpenVPN-Webgui

Abbildung 4.4.: Detailansicht einer Verbindung (Keymanagement)

Paketfilter: Zeigt die Paketfilterkonfiguration an, die für diese Verbindung gültig ist. Der Paketfilter ist nur konfiguriert, wenn die Verbindung gestartet und als Tunnel eingerichtet ist.

Bridge: Zeigt die Konfiguration der Bridges auf dem Router an. Dieser Punkt nur angezeigt, wenn die Verbindung als Bridge eingerichtet ist.

Konfiguration: Über diesen Punkt kann die beim Booten generierte Konfiguration der Verbindung angeschaut werden.

Keymanagement: Über diesen Punkt kann für diese Verbindung ein Schlüssel erzeugt und auch heruntergeladen werden. (siehe Abbildung 4.4) Ist kein Schlüssel vorhanden (beim ersten Start) wird automatisch einer generiert und angezeigt. Er kann über das Download-Symbol direkt heruntergeladen, oder mit copy/paste in eine Textdatei übertragen werden. Um einen neu generierten Schlüssel auf dem Router zu speichern, ist auf das Disketten-Symbol zu klicken. Ein solcher Speichervorgang kann mit einem Klick auf das Wiederherstellen Symbol rückgängig gemacht werden.

Supportinformationen: Hier werden alle Dinge angezeigt, die bei Problemen relevant sein könnten. Sie können diese Informationen dann beispielsweise auf Anfrage per copy&paste in einen Artikel der Newsgroup übertragen.

4.15.8. OpenVPN - Zusammenarbeit unterschiedlicher OpenVPN Versionen

Bei der Zusammenarbeit unterschiedlicher OpenVPN Versionen muss darauf geachtet werden, dass diese unterschiedliche Standardwerte für die Parameter einer Verbindung benutzen. Das betrifft insbesondere die MTU, Fragment und MSSFIX Einstellungen. Wenn die entsprechenden Werte nicht „zusammenpassen“, ist ein Verbindungsaufbau nicht möglich oder die Verbindung funktioniert zwar mit einem Pingbefehl, bricht aber beispielsweise bei der Benutzung von ssh zusammen. Typische Fehlermeldungen in einem solchen Fall sind z. B.:

```
FRAG_IN error flags=0xfa2a187b: FRAG_TEST not implemented
FRAG_IN error flags=0xfa287f34: spurrious FRAG_WHOLE flags
```

Die entscheidenden Parameter für das Zustandekommen einer Verbindung sind folgende Einstellungen:

OPENVPN_x_TUN_MTU Der MTU Wert des TUN Geräte war bei OpenVPN 1.x auf 1300 eingestellt. Ab OpenVPN 2.0 wird 1500 als Standardwert angenommen.

OPENVPN_x_LINK_MTU Die Bytegröße der Verbindung der beiden OpenVPN Daemonen. Dieser Standardwert ist abhängig von der verwendeten OpenVPN Version und des Betriebssystems.

OPENVPN_x_FRAGMENT Datenpakete (egal ob UDP oder TCP), deren Größe über der Fragmentgrenze liegt, werden auf Datenpakete aufgeteilt, die nicht größer sind als die unter `OPENVPN_x_FRAGMENT` angegeben Bytegröße.

OPENVPN_x_MSSFIX Damit TCP Verbindungen, die über das VPN Daten austauschen, nach Möglichkeit die Datenpakete nicht fragmentieren müssen, kann hier eine gewünschte

maximale Größe der TCP Datenpakete vorgegeben werden. An diese Vorgabe halten sich dann aktuelle Betriebssysteme und ein aufwendiges fragmentieren der Datenpakete ist nicht notwendig.

Die unterschiedlichen OpenVPN Versionen benutzen folgende Werte als Standardwerte. Diese Werte müssen Sie beachten, wenn Sie mit OpenVPN Versionen Kontakt aufnehmen wollen, die nicht auf einem fli4l-Router laufen. Die Standardwerte auf dem fli4l-Router sind in der zweiten Tabelle aufgeführt.

OpenVPN Version/Option	1.xx	2.00
OPENVPN_x_TUN_MTU	1300	1500
OPENVPN_x_TUN_MTU_EXTRA	unbekannt	32
OPENVPN_x_FRAGMENT	unbekannt	nicht konfiguriert
OPENVPN_x_MSSFIX	nicht konfiguriert	1450

Tabelle 4.11.: Unterschiedliche MTU Parameter der unterschiedlichen OpenVPN Versionen.

fli4l Version/Option	bis einschließlich 2.1.8	ab 2.1.9
OPENVPN_x_TUN_MTU	1300	1500
OPENVPN_x_TUN_MTU_EXTRA	64	32
OPENVPN_x_FRAGMENT	nicht konfiguriert	1300
OPENVPN_x_MSSFIX	nicht konfiguriert	1300

Tabelle 4.12.: Unterschiedliche MTU Parameter der fli4l-Router Versionen.

Aufgrund dieser unterschiedlichen Einstellungen, sollten Sie die für Ihr Netzwerk passenden Standardwerte ermitteln und diese dann explizit in die config/openvpn.txt schreiben. Folgende Werte sind in den meisten Fällen gute Startwerte für die ersten Tests.

```
OPENVPN_DEFAULT_TUN_MTU='1500'
OPENVPN_DEFAULT_MSSFIX='1300'
OPENVPN_DEFAULT_FRAGMENT='1300'
```

Leider gibt es für fli4l Versionen vor 2.1.9 keine Möglichkeit den „tun-mtu“ Parameter direkt zu setzen. Allerdings läßt sich dieser Parameter indirekt über den `OPENVPN_x_LINK_MTU` beeinflussen. Der tun-mtu Wert ist ca. 45 Bytes kleiner als der bei `OPENVPN_x_LINK_MTU` angegebene Wert. Um den genauen Wert zu ermitteln, hilft nur ausprobieren.

4.15.9. OpenVPN - Beispiele

Einige Beispiele verdeutlichen die Konfiguration des OpenVPN Paketes.

Beispiel - Zwei Netze mit fli4l-Routern verbinden

Im ersten Beispiel werden zwei fli4l-Router miteinander verbunden. Die Netzwerke hinter den fli4l-Routern sollen dabei Zugriff auf das jeweils andere Netzwerk erhalten. In diesem Beispiel wollen Peter und Maria ihre Netzwerke über ihre fli4l-Router miteinander verbinden. Peter benutzt als privates Netz 192.168.145.0/24 und als DynDNS Adresse 'peter.eisfair.net'. Bei Maria sieht es ähnlich aus, nur benutzt sie das Netzwerk 10.23.17.0/24 und als DynDNS Adresse 'maria.eisfair.net'. Das sich beide unbegrenzt vertrauen, erlauben sie sich gegenseitig den kompletten Zugriff auf ihre jeweiligen Netze.

OpenVPN Option	Peter	Maria
OPENVPN_1_NAME=	'maria'	'peter'
OPENVPN_1_REMOTE_HOST=	'maria.eisfair.net'	'peter.eisfair.net'
OPENVPN_1_REMOTE_PORT=	'10000'	'10001'
OPENVPN_1_LOCAL_PORT=	'10001'	'10000'
OPENVPN_1_SECRET=	'pema.secret'	'pema.secret'
OPENVPN_1_TYPE=	'tunnel'	'tunnel'
OPENVPN_1_REMOTE_VPN_IP=	'192.168.200.202'	'192.168.200.193'
OPENVPN_1_LOCAL_VPN_IP=	'192.168.200.193'	'192.168.200.202'
OPENVPN_1_ROUTE_N=	'1'	'1'
OPENVPN_1_ROUTE_1=	'10.23.17.0/24'	'192.168.145.0/24'
OPENVPN_1_PF_INPUT_N=	'1'	'1'
OPENVPN_1_PF_INPUT_1=	'ACCEPT'	'ACCEPT'
OPENVPN_1_PF_FORWARD_N=	'1'	'1'
OPENVPN_1_PF_FORWARD_1=	'ACCEPT'	'ACCEPT'

Tabelle 4.13.: OpenVPN Konfiguration mit 2 fli4l-Routern

Beispiel - Zwei Netze mit einer Bridge verbinden

Im nächsten Beispiel wird eine Bridge über eine Funkverbindung aufgebaut. Bei einer Bridge kann der Paketfilter nicht sinnvoll konfiguriert werden, da dort nur Ethernetframes weitergeleitet werden, aber nicht unbedingt IP-Pakete. Bitte immer daran denken, dass bei einer Bridgekonfiguration ein gemeinsames Netz benutzt werden muss. Und es dürfen keine IP-Adressen doppelt vergeben werden.

OpenVPN Option	Peter	Maria
OPENVPN_2_NAME	'bridge'	'bridge'
OPENVPN_2_REMOTE_HOST	'10.1.0.1'	'10.2.0.1'
OPENVPN_2_REMOTE_PORT	'10005'	'10006'
OPENVPN_2_LOCAL_HOST	'10.2.0.1'	'10.1.0.1'
OPENVPN_2_LOCAL_PORT	'10006'	'10005'
OPENVPN_2_FLOAT	'no'	'no'
OPENVPN_2_RESTART	'never'	'never'
OPENVPN_2_SECRET	'bridge.secret'	'bridge.secret'
OPENVPN_2_TYPE	'bridge'	'bridge'
OPENVPN_2_BRIDGE	'pema-br'	'pema-br'

Tabelle 4.14.: OpenVPN Konfiguration mit 2 fli4l-Routern deren Netzwerk über eine Funkverbindung gebridgt wird.

Zusätzlich zu den Angaben für OpenVPN muss natürlich noch eine Bridge in advanced_networking konfiguriert werden und die base.txt so angepaßt werden, dass dort die Bridge und nicht eth0 als Netzwerkdevice für das interne Netzwerk benutzt wird. Hier nochmal die relevanten Auszüge aus der Konfiguration von advanced_networking und base.

Beispiel - Zugriff für einen Roadwarrior konfigurieren

Bei diesem Beispiel (Roadwarrior) wird über ein Notebook mit Windows XP und einem GPRS Zugang der Zugang zu dem LAN hinter dem fli4l-Router ermöglicht. Dazu wird auf dem Windows XP Notebook OpenVPN installiert und die entsprechende *.ovpn Datei angepaßt. Leider ist der tun/tap Treiber unter Windows nicht ganz so flexibel wie sein Unix Gegenstück. Daher müssen die Point-to-Pointadressen für die VPN IP-Adressen in einem 255.255.255.252 (oder /30) Netz liegen. Wenn der Roadwarrior nur auf Dienste im LAN hinter und auf dem

advanced_networking Option	Peter	Maria
OPT_BRIDGE_DEV	'yes'	'yes'
BRIDGE_DEV_BOOTDELAY	'no'	'no'
BRIDGE_DEV_N	'1'	'1'
BRIDGE_DEV_1_NAME	'pema-br'	'pema-br'
BRIDGE_DEV_1_DEVNAME	'br0'	'br0'
BRIDGE_DEV_1_DEV_N	'1'	'1'
BRIDGE_DEV_1_DEV_1_DEV	'eth0'	'eth0'

Tabelle 4.15.: OpenVPN Konfiguration mit 2 fli4l-Routern deren Netzwerk über eine Funkverbindung gebridgt wird. Die Konfiguration der Bridge in advanced_networking.

base Option	Peter	Maria
IP_NET_N	'1'	'1'
IP_NET_1	'192.168.193.254/24'	'192.168.193.1/24'
IP_NET_1_DEV	'br0'	'br0'

Tabelle 4.16.: OpenVPN Konfiguration mit 2 fli4l-Routern deren Netzwerk über eine Funkverbindung gebridgt wird. Die Konfiguration der Bridge in der Basiskonfiguration (base.txt).

fli4l-Router zugreifen soll und nicht selber angesprochen werden muss, ist die Angabe einer Route auf der fli4l Seite nicht notwendig. Der Roadwarrior kann bei Bedarf über seine virtuelle IP-Adresse (`OPENVPN_3_REMOTE_VPN_IP`) angesprochen werden. Wenn der Roadwarrior über eine feste IP-Adresse verfügt, könnte man auch alternativ eine Hostroute eintragen. Wenn der Roadwarrior z.B. die feste IP-Adresse 192.168.33.33 hat, könnte man folgendes noch in die fli4l openvpn.txt Konfigurationsdatei einfügen:

```
OPENVPN_3_ROUTE_N='1'
OPENVPN_3_ROUTE_1='192.168.33.33/32'
```

Mit der Paketfilterkonfiguration, die hier im Beispiel gezeigt wird erlauben wir wieder die komplette Kommunikation in beide Richtungen. Nur auf den fli4l-Router direkt kann der Roadwarrior nicht zugreifen. Das wäre z.B. notwendig, wenn der Roadwarrior den DNS Server auf dem fli4l-Router benutzen soll.

```
OPENVPN_3_PF_FORWARD_N='1'
OPENVPN_3_PF_FORWARD_1='ACCEPT'
```

Soll der Zugriff vom Roadwarrior auf den internen DNS Server auf dem fli4l-Router erlaubt werden, muss noch folgendes zur fli4l Konfiguration dazugeschrieben werden:

```
OPENVPN_3_PF_INPUT_N='1'
OPENVPN_3_PF_INPUT_1='if:VPNDEV:any tmpl:dns ACCEPT'
```

Beispiel - WLAN Verbindung absichern

In diesem Beispiel wird eine WLAN Verbindung mit Hilfe von OpenVPN abgesichert. Es wird davon ausgegangen, dass im fli4l-Router sowohl eine LAN als auch eine WLAN Karte verwendet wird, oder ein Accesspoint an einer zusätzlichen Netzwerkkarte im fli4l angeschlossen

OpenVPN Option fli4l-Router	roadwarrior
OPENVPN_3_NAME='roadwarrior'	remote peter.eisfair.net
OPENVPN_3_LOCAL_PORT='10011'	rport 10011
OPENVPN_3_SECRET='roadwarrior.secret'	secret roadwarrior.secret
OPENVPN_3_TYPE='tunnel'	dev tun
OPENVPN_3_REMOTE_VPN_IP='192.168.200.238'	
OPENVPN_3_LOCAL_VPN_IP='192.168.200.237'	ifconfig 192.168.200.238 192.168.200.237
OPENVPN_3_ROUTE_N='0'	
OPENVPN_3_PF_FORWARD_N='1'	
OPENVPN_3_PF_FORWARD_1='ACCEPT'	
	route 192.168.145.0 255.255.255.0
	comp-lzo
	persist-tun
	persist-key
	ping-timer-rem
	ping-restart 60
	proto udp
	tun-mtu 1500
	fragment 1300
	mssfix

Tabelle 4.17.: OpenVPN Konfiguration mit einem Windowsrechner über GPRS.

ist. Ziel soll es sein, dass ein WLAN Client ohne VPN-Verbindung lediglich Zugriff auf den VPN Port des fli4l-Routers hat. Erst nach dem erfolgreichen Verbinden mit OpenVPN, soll uneingeschränkter Austausch mit dem Kabelgebundenen LAN möglich sein. Es müssen dafür auch Änderungen am DNSMASQ DHCP Server durchgeführt werden. Außerdem wird das advanced_networking Paket benötigt. Einstellungen in base.txt: `IP_NET_1` ist dabei das kabelgebundene LAN und `IP_NET_2` das WLAN.

```
IP_NET_N='2'
IP_NET_1='192.168.3.254/24'
IP_NET_1_DEV='br0'
IP_NET_2='192.168.4.254/24'
IP_NET_2_DEV='eth2'
```

Die DHCP-Range ist nach Belieben einzustellen. Für `IP_NET_2` sind aber unbedingt folgende Einstellungen hinzuzufügen:

```
DHCP_RANGE_2_DNS_SERVER='none'
DHCP_RANGE_2_NTP_SERVER='none'
DHCP_RANGE_2_GATEWAY='none'
```

Einstellung in advanced_networking.txt:

```
OPT_BRIDGE_DEV='yes'
BRIDGE_DEV_BOOTDELAY='yes'
BRIDGE_DEV_N='1'
BRIDGE_DEV_1_NAME='br'
BRIDGE_DEV_1_DEVNAME='br0'
BRIDGE_DEV_1_DEV_N='1'
BRIDGE_DEV_1_DEV_1_DEV='eth0'
```

OpenVPN Option Router	WLAN-Client
OPENVPN_4_NAME='wlan1'	
OPENVPN_4_LOCAL_HOST='192.168.4.254'	remote 192.168.4.254
OPENVPN_4_LOCAL_PORT='20001'	rport 20001
OPENVPN_4_SECRET='wlan1.secret'	secret wlan1.secret
OPENVPN_4_TYPE='bridge'	dev tap
OPENVPN_4_BRIDGE='br'	
OPENVPN_4_RESTART='never'	
OPENVPN_4_MUTE_REPLAY_WARNINGS='yes'	
	comp-lzo
	persist-tun
	persist-key
	ping-timer-rem
	ping-restart 60
	proto udp
	tun-mtu 1500
	fragment 1300
	mssfix

Tabelle 4.18.: OpenVPN Absicherung eines WLAN.

4.15.10. Weiterführende Links zum Thema OpenVPN

Abschliessend noch einige Links, die sich mit der Konfiguration von OpenVPN beschäftigen:
http://openvpn.net
http://de.wikipedia.org/wiki/OpenVPN
http://openvpn.se/
http://www.vpnforum.de/
http://arnowelzel.de/wiki/de/fli4l/openvpn
http://wiki.freifunk.net/OpenVPN
http://w3.linux-magazine.com/issue/24/Charly.pdf
http://w3.linux-magazine.com/issue/25/WirelessLAN_Intro.pdf
http://w3.linux-magazine.com/issue/25/OpenVPN.pdf
http://www.0dx.de/linux/linuxmag/1994_2009/2002/08/sysadauf/sysadauf.html

4.16. PCMCIA - PC-Card Unterstützung

4.16.1. PCMCIA-Treiber

fli4l kann auch mit PCMCIA-Karten zusammenarbeiten. Bei `OPT_PCMCIA`='yes' werden die entsprechenden Basis-Treiber installiert. Welche konkreten Kartentreiber verwendet werden sollen, wird z.B. über `NET_DRV_x` (Seite 32) eingestellt.

PCMCIA_PCIC - PCMCIA Socket-Driver

Es kann dabei gewählt werden: 'i82365' oder 'tcic' für PCMCIA Bridges, sowie 'yenta_socket' und 'i82092' für Cardbus Bridges.

Standard-Einstellung: `PCMCIA_PCIC`='i82365'

PCMCIA_PCIC_OPTS - Optionen für den PCMCIA Socket-Driver

Standard-Einstellung: `PCMCIA_PCIC_OPTS`=''

Mögliche Einstellungen: poll_interval=n n in je 10 Millisekunden - Sinnvoller Wert: 1000 Stellt das Abfrageintervall für Kartenwechsel ein irq_list=x,y,z,... Eine Liste der zu verwendenden Interrupts

PCMCIA_PCIC_EXTERN - Externes PCMCIA-Subsystem verwenden

Standard-Einstellung: `PCMCIA_PCIC_EXTERN='no'`

Mittels 'yes' kann auf das alte PCMCIA-Subsystem zurückgegriffen werden, welches in älteren Fli4l-Versionen verwendet wurde. Manche PCMCIA-Kontroller benötigen diese Einstellung. Es kommt teilweise zu Systemhängern oder es wird einfach kein PCMCIA-Controller erkannt. In diesen Fällen bitte mit 'yes' nochmals probieren.

Wenn hier 'yes' ausgewählt ist, ist nur noch der PCIC-Treiber 'i82365' möglich, da die anderen nicht betroffen sind.

Diese Variable ist optional und muß ggf. in der pcmcia.txt von Hand eingefügt werden.

PCMCIA_CORE_OPTS - Optionen für PCMCIA Core-Driver

Standard-Einstellung: `PCMCIA_CORE_OPTS=''`

PCMCIA_CARDMGR_OPTS - Optionen für PCMCIA Card-Manager

Standard-Einstellung: `PCMCIA_CARDMGR_OPTS=''`

PCMCIA_MISC_N PCMCIA_MISC_x Anzahl der zusätzlich zu ladenden PCMCIA-Module: serial_cs für Modems und Combo-Karten parport_cs Druckerschnittstellen

4.17. PPP - Anbindung eines Rechners über serielle Schnittstelle

Mit der Einstellung `OPT_PPP='yes'` ist es möglich, einen weiteren PC über die serielle Schnittstelle anzubinden. Dieses kann zum Beispiel dann sinnvoll sein, wenn man ein Notebook, welches keine Netzwerkkarte hat, in das Netzwerk einbeziehen will. Im folgenden Text wird der über die serielle Schnittstelle angeschlossene Rechner der Client-PC genannt.

PPP_DEV Hier ist der serielle Port von fli4l anzugeben. Folgende Werte sind erlaubt:

'com1' COM1-Port (klein geschrieben!)
'com2' COM2-Port (klein geschrieben!)

PPP_SPEED Hier muss die Übertragungsrate (bit/sec) eingetragen werden. 38400 wird auch von alten Schnittstellen unterstützt. Eventuell kann es Probleme geben, wenn man die Rate auf 57600 oder gar 115200 bit/s einstellt.

```
Beispiel: PPP_SPEED='38400'
```

PPP_IPADDR PPP_PEER In `PPP_IPADDR` ist die IP-Adresse von fli4l auf COM-Port-Seite einzutragen, z.B. '192.168.4.1'. In Variable `PPP_PEER` wird die zu verwendende IP-Adresse des Client-PCs eingetragen, z.B. '192.168.4.2'.

PPP_NETWORK PPP_NETMASK In PPP_NETWORK ist das verwendete Netzwerk einzutragen und in Variable PPP_NETMASK die verwendete Netzwerkmaske. Diese beide Variablen werden vom Zusatzpaket 'samba_lpd' verwendet.

Wichtig: *Dabei ist folgendes zu beachten:*

1. Die IP-Adressen dürfen **nicht** aus dem Netzwerk-Adressbereich des Ethernet-LANs stammen, sondern es muss für die Point-To-Point-Konfiguration ein eigener Netzwerk-Adressbereich verwendet werden!
2. Damit der Client-PC auch eine Verbindung zum Internet aufnehmen kann, ist das Mini-PPP-Netzwerk ebenso zu maskieren wie das LAN. Die Netzwerknummer muss also auch in MASQ_NETWORK (Seite 42) (s. nächster Abschnitt) zusätzlich eingetragen werden.
3. Man sollte den Client-PC mit in die Host-Tabelle für den DNS-Server aufnehmen.

Das hat folgenden Grund:

Möchte man vom Client-PC telnet oder ftp zum fli4l-Router verwenden, machen die entsprechenden Daemons auf fli4l einen Reverse-DNS-Lookup, um festzustellen, wer denn da eine Verbindung wünscht. Ist der Client-PC nicht in der Host-Tabelle eingetragen, wird eine Verbindung in's Internet hergestellt, um den Namen des Clients herauszufinden. Und dieses kann durch Eintragen des Client-PCs in der Host-Tabelle vermeiden.

Beispiel für eine PPP-Konfiguration über die serielle Schnittstelle:

```
PPP_DEV='com1'
PPP_SPEED='38400'
PPP_IPADDR='192.168.4.1'
PPP_PEER='192.168.4.2'
PPP_NETWORK='192.168.4.0'
PPP_NETMASK='255.255.255.0'
```

und weiter in config/base.txt:

```
MASQ_NETWORK='192.168.6.0/24 192.168.4.0/24'
```

Dabei ist die erste Netzwerknummer die des Ethernet-LANs, die zweite die des PPP-Netzwerks.

Solange nicht über ISDN auf ein weiteres 192.168.er Netz zugegriffen wird (ist bei mir jedoch der Fall - wegen zusätzlichem Anschluß an Firma, s.u.) sollte man beide Netzwerke in MASQ_NETWORK (Seite 42) zusammenfassen. Dies vereinfacht nämlich die Firewall-Regeln.

Daher besser:

```
MASQ_NETWORK='192.168.0.0/16'
```

Das heißt: "Maskiere alles, was mit 192.168. beginnt".

Als letztes ist noch die DNS-Konfiguration anzupassen, z.B.:

```
HOST_5='192.168.4.2 serial-pc'
```

Nicht vergessen, HOSTS_N (Seite 101) zu inkrementieren!

Ist der Client-PC ein Windows-Rechner, kann dort über die Konfiguration des DFÜ-Adapters eine PPP-Verbindung zum fli4l-Router konfiguriert werden.

Bei Verwendung eines Linux-Rechners erstellt man am besten folgendes Shellscript auf dem Client-PC (z.B. /usr/local/bin/ppp-on):

```
#! /bin/sh
dev='/dev/ttyS0'                    # COM1, für COM2: ttyS1
speed='38400'                       # Geschwindigkeit
options='defaultroute crtscts'      # Optionen
myip='192.168.4.2'                  # IP-Adresse Notebook
fli4lip='192.168.4.1'               # IP-Adresse fli4l-Router
pppd $dev $speed $options $myip:$fli4lip &
```

Sollte es es damit Probleme geben: man pppd

Auch muss der fli4l-Rechner als DNS-Server auf dem Client-PC eingetragen werden, wenn man mit dem Rechner eine Verbindung zum Internet wünscht. Es müssen dafür in /etc/resolv.conf des Client-PCs folgende zwei Zeilen eingetragen werden: die gewählte Domain und die Ethernet-IP-Adresse des fli4l-Router als Nameserver.

Beispiel:

```
search domain.de
nameserver 192.168.1.4
```

"domain.de" bzw. "192.168.1.4" sind durch die entsprechenden Werte zu ersetzen. Wichtig: Die IP-Adresse muß die der Ethernet-Karte des fli4l-Routers sein!

Als serielle Verbindung wird ein sogenanntes Nullmodemkabel (Seite 349) verwendet. Die Anschluß-Belegung ist im Anhang der Base-Dokumentation beschrieben.

Ein Howto, wie ein Windows-Client über serielles PPP angebunden werden kann, hat Oliver Walter verfasst:

http://www.fli4l.de/hilfe/howtos/basteleien/opt-ppp-howto/

4.18. PROXY - Verschiedene Proxy-Server

4.18.1. OPT_PRIVOXY - Ein Werbung-filternder HTTP-Proxy

Privoxy wird auf der offiziellen Privoxy-Homepage (http://www.privoxy.org/) als "Privacy Enhancing Proxy" (="privatsphärenerweiternder Proxy") beworben. Als sichbarer und erwünschenwerter Nebeneffekt ersetzt Privoxy Werbe-Banner und -Popups durch leere Bilder, verhindert das Speichern von ungewollten Cookies (kleine Datenpakete, mit denen eine Website einen bestimmten Surfer wiedererkennen kann) und verhindert die Anzeige von sogenannten Web-Bugs (das sind 1x1 Pixel große Bilder, die dazu benutzt werden, um das Surfverhalten von Benutzern auszuspähen).

Privoxy kann, während es läuft, ganz einfach über ein Webinterface konfiguriert und (de)aktiviert werden. Dieses Webinterface findet sich unter http://config.privoxy.org/ oder http://p.p/.

Eine so getätige Konfiguration überlebt keinen Neustart... (tobi

Privoxy ist eine konsequente Weiterentwicklung des Internet Junkbusters, der bis Version 2.1.0 in diesem Paket (http://www.junkbuster.com/) enthalten war. Die wichtigste Neuerung ist, dass alle Regeln für die Filterung in der zentralen Datei `default.action` definiert werden. Diese befindet sich bei FLI4L im Verzeichnis `/etc/privoxy`. Der große Vorteil an dieser Methode ist, dass sich neue Versionen dieser Datei seperat von http://sourceforge.net/projects/ijbswa/files/ herunterladen lassen. So kann jeder FLI4L-Benutzer die Datei selbst auf dem neusten Stand halten, ohne auf Updates von FLI4L angewiesen zu sein. (Momentan befindet sich die Version 1.8 dieser Datei im Paket.)

PRIVOXY_MENU Fügt dem httpd-Menü einen Privoxy-Abschnitt hinzu.

PRIVOXY_N Gibt die Anzahl der Privoxy-Instanzen an, die gestartet werden sollen.

PRIVOXY_x_LISTEN Hier werden die IP-Adressen oder symbolischen Namen inklusive der Portnummer der Interfaces angegeben, auf denen Privoxy auf Verbindungen von Clients horchen soll. Es ist eine gute Idee, hier nur die Adressen der Interfaces anzugeben, denen man vertraut, da alle Rechner vollen Zugriff auf Privoxy haben (und auf den eventuell aktivierten Konfigurations-Editor). In der Regel ist die Vorgabe `IP_NET_1_IPADDR:8118` sinnvoll.

Auf hier angegebenen Adressen lauscht Privoxy und bietet seine Dienste an. 8118 ist der Standard-Port. Die Angabe hier muss man dann bei der Proxy-Konfiguration des jeweils verwendeten Internet-Browsers benutzen. Weitere Informationen zur Konfiguration von Internet Explorer und Netscape Navigator auf

http://www.privoxy.org/

Genaue URL

Als Proxy beim jeweiligen Browser muss der fli4l-Rechner angegeben werden, also das, was man bei HOSTNAME='fli4l' angegeben hat bzw. dessen IP (z.B 192.168.6.1), die man bei `HOST_x_IP`='192.168.6.1' angegeben hat. Zusammen mit dieser Port-Angabe hier hat man dann alle nötigen Daten, um seinen Webbrowser für die Nutzung von Privoxy zu konfigurieren.

PRIVOXY_x_ALLOW_N Gibt die Anzahl der Listeneinträge an.

PRIVOXY_x_ALLOW_x Die Liste der Netze und/oder IP-Adressen für die der Paketfilter geöffnet wird. Sinnvoll ist hier auch wieder die Vorgabe `IP_NET_1`.

PRIVOXY_x_ACTIONDIR Diese Variable gibt den Ort an, an dem die Privoxy-Regelsätze (die Dateien *default.action* und *user.action*) auf dem Router liegen sollen. Der angegebene Pfad wird relativ zum Wurzelverzeichnis ausgewertet. Diese Variable kann für zwei Dinge verwendet werden:

Verlagern der Standardregeln auf permanenten Speicher Gibt man als Verzeichnis einen Ort ausserhalb der Ram-Disk an, werden die Standardregelsätze beim erstmaligen Booten dorthin kopiert und dann von diesem Ort aus genutzt. Änderungen an diesen Regelsätzen überleben dann ein Reboot des Routers. Zu beachten ist, dass nach einem Update des Privoxy-Paketes diese Regeln immer noch Verwendung

finden und evtl. mit dem aktuellen Paket kommende neuere Regelsätze ignoriert werden.

Verwenden eigener Regelsätze Fli4l gestattet das Überschreiben der Standardregeln mit nutzerspezifischen Regeln. Dazu legt man im *config*-Verzeichnis ein Unterverzeichnis an (z.B. *etc/my_privoxy*; es darf nicht *etc/privoxy* heissen) und legt dort die eigenen Regeln ab.

Das Setzen dieser Variable ist optional.

PRIVOXY_x_HTTP_PROXY Möchte man zusätzlich zu Privoxy einen weiteren HTTP Proxy verwenden, der dann z.B. auch Webseiten zwischenspeichert, so kann man den hier angeben. Privoxy bedient sich dann dieses Proxys. So kann man die Vorteile mehrerer Proxys nutzen. Ein Eintrag könnte so aussehen:

```
PRIVOXY_1_HTTP_PROXY='mein.provider.de:8000'
```

Die Angabe ist optional.

PRIVOXY_x_SOCKS_PROXY Möchte man zusätzlich zu Privoxy einen weiteren SOCKS Proxy verwenden, kann man den hier angeben. Um die Privatspähre weiter zu erhöhen kann der Datenverkehr vom Privoxy beispielsweise durch das Tor Netzwerk geschickt werden. Für weitere Details zu Tor lesen Sie in der Tor Dokumentation (Seite 216) weiter. Ein Eintrag für dei Nutzung von Tor könnte so aussehen:

```
PRIVOXY_x_SOCKS_PROXY='127.0.0.1:9050'
```

Die Angabe ist optional.

PRIVOXY_x_TOGGLE Diese Option aktiviert die Möglichkeit, den Proxy über das Webinterface ein- bzw. auszuschalten. Wird Privoxy ausgeschaltet, wirkt er als einfacher Forwarding-Proxy und ändert den Inhalt der übertragenen Seiten in keinster Weise. Es ist zu beachten, daß diese Einstellung für ALLE Benutzer des Proxys gilt, d.h. wenn ein Benutzer Privoxy abschaltet, ist Privoxy auch für die anderen Nutzer nur noch ein Weiterleitungs-Proxy.

PRIVOXY_x_CONFIG Diese Option ermöglicht den Benutzern des Proxys die interaktive Bearbeitung der Konfiguration über das Privoxy-Webinterface. Für weitere Details bitte ich auch hier, die Privoxy-Dokumentation zu konsultieren.

PRIVOXY_x_LOGDIR Mit dieser Option kann ein Logverzeichnis für Privoxy angegeben werden. Dies kann z.B. dann sinnvoll sein, wenn Website-Zugriffe der Benutzer geloggt werden sollen. Wird hier nichts angegeben (Standard), dann loggt nur die wichtigsten Meldungen auf die Konsole und ignoriert `PRIVOXY_LOGLEVEL`.

PRIVOXY_x_LOGLEVEL Diese Option gibt an, was Privoxy in die Logdatei schreiben soll. Folgende Werte sind möglich, sie können durch Leerstelle getrennt angegeben werden, man kann sie aber auch addieren.

Wert	Was wird geloggt?
1	Jeden Request (GET/POST/CONNECT) ausgeben.
2	Status jeder Verbindung ausgeben
4	I/O-Status anzeigen
8	Header-Parsing anzeigen
16	**Alle** Daten loggen
32	Force-Feature debuggen
64	reguläre Ausdrücke debuggen
128	schnelle Weiterleitungen debuggen
256	GIF De-Animation debuggen
512	Common Log Format (zur Logfile-Analyse)
1024	Popup-Kill-Funktion debuggen
2048	Zugriffe auf den eingebauten Webserver loggen
4096	Startmeldungen und Warnungen
8192	Nicht-fatale Fehler

Um eine Logdatei im Common Logfile Format zu erstellen, sollte NUR der Wert 512 angegeben werden, da sonst die Logdatei durch andere Meldungen "verschmutzt" wird und somit nicht mehr problemlos ausgewertet werden kann.

Privoxy bietet sehr viele Konfigurationsmöglichkeiten. Diese können aber aus verständlichen Gründen nicht alle durch die Konfigurations-Datei von fli4l abgedeckt werden. Sehr viele dieser Optionen können im Webinterface von Privoxy eingestellt werden. Genauere Infos über den Aufbau dieser Dateien gibt es auf der Privoxy-Homepage. Die Konfigurationsdateien von Privoxy liegen unter <FLI4L-Verzeichnis>/opt/etc/privoxy/. Es handelt sich hierbei um die Orginal-Dateien aus dem Privoxy-Paket, allerdings wurden, um Platz zu sparen, alle Kommentare entfernt.

4.18.2. OPT_TOR - Ein anonymes Kommunikationssystem für das Internet

Tor ist ein Werkzeug für eine Vielzahl von Organisationen und Menschen, die ihren Schutz und ihre Sicherheit im Internet verbessern wollen. Die Nutzung von Tor hilft Ihnen, das Browsen und Veröffentlichen im Web, Instantmessaging, IRC, SSH und anderen TCP basierende Anwendungen zu anonymisieren. Weiterhin bietet Tor eine Plattform auf der Softwareentwickler neue Anwendungen schaffen können die zu mehr Anonymität, Sicherheit und zum Schutz der Privatsphäre beitragen.

https://www.torproject.org/index.html.de

TOR_LISTEN_N

TOR_LISTEN_x Hier werden die IP-Adressen oder symbolischen Namen inklusive der Portnummer der Interfaces angegeben, auf denen Tor auf Verbindungen von Clients horchen soll. Es ist eine gute Idee, hier nur die Adressen der Interfaces anzugeben, denen man vertraut, da alle Rechner vollen Zugriff auf Tor haben (und auf den eventuell aktivierten Konfigurations-Editor). In der Regel ist die Vorgabe `IP_NET_1_IPADDR:9050` sinnvoll.

Auf hier angegebenen Adressen lauscht Tor und bietet seine Dienste an. 9050 ist der Standard-Port. Die Angabe hier muss man dann bei der Proxy-Konfiguration des jeweils verwendeten Programms benutzen.

Als Proxy beim jeweiligen Programm muss der fli4l-Rechner angegeben werden, also das, was man bei HOSTNAME='fli4l' angegeben hat bzw. dessen IP (z.B 192.168.6.1), die man bei HOST_x_IP='192.168.6.1' angegeben hat. Zusammen mit dieser Port-Angabe hier hat man dann alle nötigen Daten, um sein Programm für die Nutzung von Tor zu konfigurieren.

TOR_ALLOW_N Gibt die Anzahl der Listeneinträge an.

TOR_ALLOW_x Die Liste der Netze und/oder IP-Adressen für die der Paketfilter geöffnet wird. Sinnvoll ist hier auch wieder die Vorgabe IP_NET_1.

TOR_CONTROL_PORT Hier kann angegeben werden auf welchem TCP Port Tor einen Kontrollzugang über das Tor Control Protocol öffnen soll. Die Angabe ist optional. Wird nichts angegeben wird diese Funktion deaktiviert.

TOR_CONTROL_PASSWORD Hier kann ein Passwort für den Kontrollzugang angegeben werden.

TOR_DATA_DIR Diese Angabe ist optional. Wird nichts angegeben, wird der Standardordner /etc/tor verwendet

TOR_HTTP_PROXY Soll Tor die Anfragen an einen HTTP-Proxy weiterleiten, kann man den hier angeben. Tor bedient sich dann dieses Proxys. So kann man die Vorteile mehrerer Proxys nutzen. Ein Eintrag könnte so aussehen:

```
TOR_HTTP_PROXY='mein.provider.de:8000'
```

Die Angabe ist optional.

TOR_HTTP_PROXY_AUTH Eine eventuell notwendige Authentifizierung für den Proxy kann hier in der Form Benutzername:Passwort eingetragen werden.

TOR_HTTPS_PROXY Hier kann ein HTTPS-Proxy eingetragen werden. Siehe dazu auch TOR_HTTP_PROXY.

TOR_HTTPS_PROXY_AUTH Siehe dazu TOR_HTTP_PROXY_AUTH.

TOR_LOGLEVEL Diese Option gibt an, was Tor in die Logdatei schreiben soll. Folgende Werte sind möglich: debug, info, notice, warn oder err Die Werte debug und info sollten aus Sicherheitsgründen möglichst nicht verwendet werden.

TOR_LOGFILE Falls Tor statt ins syslog in eine Datei loggen soll, kann diese hier angegeben werden.

4.18.3. OPT_SS5 - Ein Socks4/5 Proxy

Für einige Programm wird ein Socks-Proxy benötigt. SS5 stellt diese Funktionalität bereit.
http://ss5.sourceforge.net/

SS5_LISTEN_N

SS5_LISTEN_x Hier werden die IP-Adressen oder symbolischen Namen inklusive der Portnummer der Interfaces angegeben, auf denen SS5 auf Verbindungen von Clients horchen soll. Es ist eine gute Idee, hier nur die Adressen der Interfaces anzugeben, denen man vertraut, da alle Rechner vollen Zugriff auf SS5 haben (und auf den eventuell aktivierten Konfigurations-Editor). In der Regel ist die Vorgabe `IP_NET_1_IPADDR:8050` sinnvoll.

Auf hier angegebenen Adressen lauscht SS5 und bietet seine Dienste an. 8050 ist der Standard-Port. Die Angabe hier muss man dann bei der Proxy-Konfiguration des jeweils verwendeten Programms benutzen.

Als Proxy beim jeweiligen Programm muss der fli4l-Rechner angegeben werden, also das, was man bei HOSTNAME='fli4l' angegeben hat bzw. dessen IP (z.B 192.168.6.1), die man bei `HOST_x_IP`='192.168.6.1' angegeben hat. Zusammen mit dieser Port-Angabe hier hat man dann alle nötigen Daten, um sein Programm für die Nutzung von SS5 zu konfigurieren.

SS5_ALLOW_N Gibt die Anzahl der Listeneinträge an.

SS5_ALLOW_x Die Liste der Netze und/oder IP-Adressen für die der Paketfilter geöffnet wird. Sinnvoll ist hier auch wieder die Vorgabe `IP_NET_1`.

4.18.4. OPT_TRANSPROXY (EXPERIMENTELL) - Transparanter HTTP-Proxy

Transproxy ist ein „transparenter" Proxy, also ein Programm, dass es ermöglicht, alle HTTP-Abfragen, die über den Router laufen, abzufangen und an einen normalen HTTP-Proxy, z.B. Privoxy, weiterzuleiten. Um diese Funktionalität zu erreichen, muss der Paketfilter HTTP-Anfragen, die eigentlich ins Internet gehen sollen, an Transproxy weiterreichen, welcher diese weiter aufbereitet und an den anderen HTTP-Proxy weitergibt. iptables bietet zur Unterstützung dieser Funktion die Aktion „REDIRECT":

```
PF_PREROUTING_1='tmpl:http IP_NET_1 REDIRECT:8081'
```

Diese Regel würde alle HTTP-Pakete aus dem ersten definierten Netz (normalerweise das interne LAN) an Transproxy auf Port 8081 weiterleiten.

TRANSPROXY_LISTEN_N

TRANSPROXY_LISTEN_x Hier werden die IP-Adressen oder symbolischen Namen inklusive der Portnummer der Interfaces angegeben, auf denen Transproxy auf Verbindungen von Clients horchen soll. Hier müssen alle Interfaces angegeben werden, für die im Paketfilter Pakete auf Transproxy umgelenkt werden. Mit der Vorgabeeinstellung `any:8081` hört Transproxy auf allen Interfaces.

TRANSPROXY_TARGET_IP

TRANSPROXY_TARGET_PORT Mit diesen Optionen wird festgelegt, an welchen Dienst eingehende HTTP-Anfragen umgeleitet werden. Dies kann ein beliebiger Standard-HTTP-Proxy (Squid, Privoxy, Apache, etc.) auf einem beliebigen anderen Rechner (oder auch auf fli4l selbst) sein. Hier ist darauf zu achten, dass der Proxy sich nicht im Bereich der durch den Paketfilter umgeleiteten HTTP-Anfragen befindet, da sonst eine Schleife entsteht.

TRANSPROXY_ALLOW_N

TRANSPROXY_ALLOW_x Die Liste der Netze und/oder IP-Adressen für die der Paketfilter geöffnet wird. Dies sollte die gleichen Netze abdecken, die auch im Paketfilter umgeleitet werden. Werden hier keine Bereiche angegeben, müssen die Angaben von Hand in der Paketfilter-Konfiguation vorgenommen werden.

4.19. QoS - Quality of Service

Mit QoS kann man die verfügbare Bandbreite regulieren und zum Beispiel auf verschiedene Ports, IP's und noch einiges mehr zu verteilen.

Ein Modem verwaltet eine Packet-Queue (Queue = Schlange, Reihe (in einer Schlange stehen)) in der Pakete gespeichert werden, die die verfügbare Bandbreite überschreiten. Bei DSL-Modems zum Beispiel, sind diese sehr groß. Das hat den Vorteil, daß recht gleichmäßig die maximale Bandbreite ausgenutzt werden kann. Denn schickt der Router an das Modem für kurze Zeit (sehr kurz) weniger Pakete, dann hat das Modem noch immer Pakete in der Queue, die es abzuarbeiten gilt. So eine Queue ist sehr simpel gehalten, denn dort geht alles der Reihe nach, ist eben ein faires Modem :-D

Und hier kommt dann QoS ins Spiel. QoS verwaltet auch eine Packet-Queue, nur eben im Router selber und dort hat man die Möglichkeit König zu sein, eben zu entscheiden welches Paket zu erst darf und welche Pakete sich noch ein bischen zurückhalten müssen. Wenn alles richtig konfiguriert wurde, dann sendet QoS die Pakete gerade eben so schnell, daß sie nicht in die Packet-Queue des Modems landen. Das wäre so, als hätte man die Queue vom Modem in den Router geholt.

Noch etwas allgemeines zu den Geschwindigkeitseinheiten: QoS unterstützt Mibit/s (mebibit/s) und Kibit/s (kibibit/s), wobei gilt 1Mibit = 1024Kibit.

4.19.1. Konfiguration

OPT_QOS Hier ist yes zusetzen wenn man das `OPT_QOS` einsetzen will und no wenn man das Gegenteil beabsichtigt

QOS_INTERNET_DEV_N Die Anzahl der Devices, die Daten ins Internet routen.

QOS_INTERNET_DEV_x Hier sollte die Liste der Devices eingetragen werden, über die Daten ins Internet übertragen werden. Beispiele:

`QOS_INTERNET_DEV_N='3'`	Anzahl der Geräte
`QOS_INTERNET_DEV_1='ethX'`	für Kabel und sonstige Ethernet-Verbindungen
`QOS_INTERNET_DEV_2='ppp0'`	für DSL über PPPoE
`QOS_INTERNET_DEV_3='ipppX'`	für ISDN

Das ISDN-Device für den ersten Circuit dürfte das ippp0 lauten, für den 2. ippp1. Wenn aber für den ersten Circuit Kanalbündelung aktiviert wurde, dann heißt der 2. Kanal des 1. Circuit ippp1 und der 2. Circuit ippp2. Man sollte QOS mit ISDN nur dann nutzen, wenn Kanalbündelung für den benutzten Circuit deaktiviert ist.

QOS_INTERNET_BAND_DOWN Maximale Downstreambandbreite des Internetzugangs. Siehe weiter oben: Hinweis zu den Geschwindigkeitseinheiten (Seite 219).

Hinweis: Für zeitkritische Aufgaben, wie das bevorzugen von ACK-Paketen, ist es nötig die Bandbreite nicht höher zu setzen als wirklich vorhanden, da man sonst zwar innerhalb der Packet-Queue auf dem Router die Pakete sortiert, dies dann aber nicht ganz korrekt gemacht wird und letztendlich doch wieder in der Packet-Queue des Modems aufgehalten werden. Möglich ist es außerdem, das die vom Provider angegebene Bandbreite nicht hundertprozentig mit der wirklich verfügbaren übereinstimmt, es könnte ein bischen mehr oder auch weniger sein. Da ist also ausprobieren angesagt.

QOS_INTERNET_BAND_UP Maximale Upstreambandbreite des Internet-Zugangs. Siehe Hinweis zu den Geschwindigkeitseinheiten unter `OPT_QOS`.

Hinweis: Siehe Hinweis bei `QOS_INTERNET_BAND_DOWN`.

QOS_INTERNET_DEFAULT_DOWN Hier ist die Standardklasse für Pakete anzugeben, die aus dem Internet kommen. Alle Pakete, die nicht durch einen Filter in eine Klasse gesteckt wurden, landen dann in der angegebenen Klasse.

Wurde keine Klasse eingerichtet, für die die Variable

```
QOS_CLASS_x_DIRECTION='down'
```

gesetzt wurde, so setzt man:

```
QOS_INTERNET_DEFAULT_DOWN='0'
```

Beispiel:

Es wurden 2 Klassen eingerichtet und ein Filter steckt alle Pakete, die z.B. an eine bestimmte IP-Adresse geschickt wurden in die 1. von den beiden. Alle anderen Pakete sollen in die 2. Klasse gesteckt werden. Folglich müßte hier

```
QOS_INTERNET_DEFAULT_DOWN='2'
```

eingetragen werden.

Es ist darauf zu achten, daß für `QOS_INTERNET_DEFAULT_DOWN` eine Klasse angegeben wird, deren `QOS_CLASS_x_DIRECTION` Variable das Argument down enthält.

QOS_INTERNET_DEFAULT_UP Hier ist die Standardklasse für Pakete anzugeben, die in das Internet gehen. Alle Pakete, die nicht durch einen Filter in eine Klasse gesteckt wurden, landen dann in der angegebenen Klasse.

Wurde keine Klasse eingerichtet, für die die Variable

```
QOS_CLASS_x_DIRECTION='up'
```

gesetzt wurde, so setzt man:

```
QOS_INTERNET_DEFAULT_UP='0'
```

Das ganze funktioniert analog zu `QOS_INTERNET_DEFAULT_DOWN`.

Es ist darauf zu achten, daß für `QOS_INTERNET_DEFAULT_UP` eine Klasse angegeben wird, deren `QOS_CLASS_x_DIRECTION` Variable das Argument up enthält.

QOS_CLASS_N Hier ist die gewünschte Anzahl der Klassen (engl. Class) anzugeben.

QOS_CLASS_x_PARENT Mit dieser Variable kann man Klassen verschachteln. Man gibt hier immer die Nummer der Vaterklasse an. Die Bandbreite die der Vaterklasse zugeteilt wurde, kann dann unter den Unterklassen weiter aufgeteilt werden. Die maximale Verschachtelungstiefe beträgt hier 8 Ebenen, wobei das Interface selber schon eine Ebene darstellt, es bleiben also maximal 7 konfigurierbar.

Soll die Klasse keine Unterklasse sein, so gibt man hier folgendes an:

```
QOS_CLASS_x_PARENT='0'
```

Ihr wird dann je nachdem zu welcher Richtung sie gehört (sie QOS_CLASS_x_PORT_TYPE), maximal die in QOS_CLASS_x_PORT_TYPE oder QOS_INTERNET_BAND_DOWN angegebene Bandbreite zugeteilt.

Wichtig: Falls hier nicht '0' angegeben wird, so ist darauf zu achten, daß angegebene Klasse vor der zu der diese Variable gehört steht (auf die Nummerierung bezogen).

QOS_CLASS_x_MINBANDWIDTH Bandbreite, die man der Klasse zusprechen will. Man könnte hier auch von einem Verhältnis sprechen. Siehe Hinweis zu den Geschwindigkeitseinheiten unter OPT_QOS.

Beispiel: Man hat eine Klasse, dessen Bandbreite auf 128Kibit/s beschränkt ist.:

```
QOS_CLASS_1_MINBANDWIDTH='128Kibit/s'
QOS_CLASS_1_MAXBANDWIDTH='128Kibit/s'
QOS_CLASS_1_PARENT='0'
```

Weiterhin hat man 3 Klassen dessen QOS_CLASS_x_MINBANDWIDTH- und QOS_CLASS_x_-MAXBANDWIDTH-Einstellungen wie folgt aussehen und alle Unterklassen unserer ersten Klasse sind:

```
QOS_CLASS_2_PARENT='1'
QOS_CLASS_2_MINBANDWIDTH='60Kibit/s'
QOS_CLASS_2_MAXBANDWIDTH='128Kibit/s'

QOS_CLASS_3_PARENT='1'
QOS_CLASS_3_MINBANDWIDTH='40Kibit/s'
QOS_CLASS_3_MAXBANDWIDTH='128Kibit/s'

QOS_CLASS_4_PARENT='1'
QOS_CLASS_4_MINBANDWIDTH='28Kibit/s'
QOS_CLASS_4_MAXBANDWIDTH='128Kibit/s'
```

Alle Unterklassen besitzen die selbe (oder auch keine) Priorität (siehe QOS_CLASS_x_-PRIO). Wird nun auf jede dieser 3 Klassen mehr Verkehr produziert als in ihrer jeweiligen QOS_CLASS_x_MINBANDWIDTH angegeben, so bekommt jede Klasse entsprechend ihrer QOS_-CLASS_x_MINBANDWIDTH-Einstellung Bandbreite zugewiesen. Wenn aber z.B. Klasse 2 nur 20Kibit/s an Verkehr produziert, dann läßt dies Klasse ja 40Kibit/s "übrig". Dieser Überschuß wird im Verhältnis 40/28 unter Klasse 3 und 4 aufgeteilt. Jede Klasse selber ist

durch QOS_CLASS_x_MAXBANDWIDTH auf 128Kibit/s beschränkt und da sie alle Unterklassen einer auf 128Kibit/s beschränkten Klasse sind, können sie auch alle zusammen nicht mehr als 128Kibit/s konsumieren.

QOS_CLASS_x_MAXBANDWIDTH Bandbreite, die man der Klasse maximal zuteilen will. Es macht keinen Sinn einen niedrigeren Wert als der in QOS_CLASS_x_MINBANDWIDTH einzutragen. Gibt man hier nichts an, so nimmt diese Variable automatisch den Wert von QOS_CLASS_x_MINBANDWIDTH an. Eine solche Klasse kann dann natürlich keine überschüssige Bandbreite beanspruchen.

Siehe Hinweis zu den Geschwindigkeitseinheiten unter OPT_QOS.

QOS_CLASS_x_DIRECTION Mit dieser Variable wird angegeben, zu welcher Richtung die Klasse gehört. Soll sie zur Regulierung des Upstreams benutzt werden, so ist hier

```
QOS_CLASS_x_DIRECTION='up'
```

anzugeben, für den Dwonstream analog:

```
QOS_CLASS_x_DIRECTION='down'
```

QOS_CLASS_x_PRIO Hier wird geregelt, welche Priorität ein Klasse hat. Je niedriger die Nummer, desto höher die Priorität. Erlaubt sind werte zwischen 0 und 7. Wenn die Variable leer gelassen wird, so kommt das dem setzen einer 0 gleich.

Wenn eine Priorität gesetzt wird, dann wird darüber bestimmt, welcher Klasse zuerst Überschüssige Bandbreite angeboten wird. Um das klar zu machen, ändern wir das Beispiel aus QOS_CLASS_x_MINIMUMBANDWIDTH leicht ab: An der ersten Klasse wird nichts Verändert. Die Klassen 2-4 bekommen eine Priorität zugewiesen:

```
QOS_CLASS_2_PARENT='1'
QOS_CLASS_2_MINBANDWIDTH='60Kibit/s'
QOS_CLASS_2_MAXBANDWIDTH='128Kibit/s'
QOS_CLASS_2_PRIO='1'

QOS_CLASS_3_MINBANDWIDTH='40Kibit/s'
QOS_CLASS_3_PARENT='1'
QOS_CLASS_3_MAXBANDWIDTH='128Kibit/s'
QOS_CLASS_3_PRIO='1'

QOS_CLASS_4_PARENT='1'
QOS_CLASS_4_MINBANDWIDTH='28Kibit/s'
QOS_CLASS_4_MAXBANDWIDTH='128Kibit/s'
QOS_CLASS_4_PRIO='2'
```

Wie in dem Ursprungsbeispiel konsumiert Klasse 2 nur 20Kibit/s und läßt somit einen Überschuß von 40Kibit/s übrig. Klasse 3 und 4 wollen noch immer mehr Bandbreite als überhaupt verfügbar. Da nun aber Klasse 3 eine höhere Priorität als Klasse 4 hat, darf sie den Überschuß von 40Kibit/s vertilgen.

Angenommen Klasse 3 braucht aber nur 20Kibit/s des ursprünglichen Überschusses von 40Kibit/s, dann bekommt Klasse 4 die restlichen 20Kibit/s.

Nehmen wir nochmals etwas anderes an: Klasse 4 verbraucht gar keine Bandbreite und Klasse 2 und 3 wollen mehr als es überhaupt gibt. Dann bekommt jede erstmal ihre in `QOS_CLASS_x_MINBANDWIDTH` angegebene Bandbreite und der Rest wird unter ihnen im 60/40 Verhältnis aufgeteilt, da beide Klassen die selbe Priorität haben.

Wie man also sieht beeinflußt `QOS_CLASS_x_PRIO` nur, wie ein eventueller Bandbreitenüberschuß aufgeteilt wird.

QOS_FILTER_N Gewünschte Anzahl der der Filter angeben.

Zu den Filtern allgemein läßt sich noch folgendes sagen: Die Argumente von verschiedenen Variablen sind UND-verknüpft, mehrere Argumente der selben Variable sind ODER-verknüpft. Soll heißen: Wird zum Beispiel in einem und dem selben Filter nach einer IP-Adresse und einem Port gefiltert, so werden nur Pakete herausgefiltert und in die Zielklasse(n) gesteckt, die auf beides gleichzeitig zutreffen.

Ein weiteres Beispiel: In einem und dem selben Filter sind zwei Ports (21 und 80) und eine IP-Adresse angegeben. Ein Datenpaket kann natürlich nicht von zwei Ports gleichzeitig kommen. Es verhält sich dann so: Der Filter filtert Pakete heraus, die entweder Port 21 benutzen und gleichzeitig die IP-Adresse, oder von Port 80 kommen und gleichzeitig von der IP-Adresse.

Wichtig: Es kommt auf die Reihenfolge der Filter an!

Ein Beispiel: Man möchte den Verkehr, der über den Port 456 läuft, für **alle** Rechner in Klasse A stopfen. Zusätzlich möchte man alle Pakete an den Rechner mit der IP 192.168.6.5 - bis auf die Pakete über Port 456 - in Klasse B laufen lassen. Richte ich nun den Filter mit der IP als erstes ein, dann landen alle Pakete - auch die über Port 456 laufen - in Klasse B und ein nachfolgender Filter für den Port 456 ändert auch nichts daran. Der Filter für den Port 456 muß also noch vor dem Filter mit der IP 192.168.6.5 stehen.

QOS_FILTER_x_CLASS Mit dieser Variable stellt ihr ein, in welche Klasse das Paket, auf den dieser Filter zutrifft, gesteckt werden soll. Möchte man zum Beispiel die gefilterten Pakete in die Klasse, die mit den Variabeln `QOS_CLASS_25_MINBANDWIDTH` spezifiziert wurde, stecken, so müßte man hier

```
QOS_FILTER_x_CLASS='25'
```

setzen.

Mit `QOS_CLASS_x_DIRECTION` gibt man ja an, ob eine Klasse nun zum Up- oder Downstream gehört. Wenn nun ein Filter gesetzt wird, der gefilterte Pakete zum Beispiel in eine Upstream-Klasse wandern läßt, dann werden auch nur Pakete aus dem Upstream von diesem Filter gefiltert und in die angegebene Klasse gesteckt. `QOS_CLASS_x_DIRECTION` bestimmt also in welcher "Richtung" gefiltert wird.

Seit Version 2.1 ist es nun auch möglich mehr als eine Zielklasse anzugeben. Möchte man zum Beispiel den Verkehr über Port 456 sowohl für den Upstream als auch für den Downstream klassifizieren, so würde man hier

```
QOS_FILTER_x_CLASS='4 25'
```

angeben, wobei zum Beispiel Klasse Nummer 4 die Upstreamklasse ist und 25 die Downstreamklasse. Es macht keinen Sinn hier jeweils mehr als ein Up- und Downstreamklassen anzugeben anzugeben, somit wird man auch nie mehr als zwei Zielklassen eintragen.

QOS_FILTER_x_IP_INTERN Hier können IP-Adressen und IP-Bereiche aus den internen Netzwerkwerken, nach denen gefiltert werden soll, angegeben werden. Sie sind durch Leerzeichen zu trennen und können frei kombiniert werden.

Das könnte zum Beispiel so aussehen:

```
QOS_FILTER_x_IP_INTERN='192.168.6.0/24 192.168.5.7 192.168.5.12'
```

Hier werden alle Adressen in der Form 192.168.6.X gefiltert und zusätzlich noch die IPs 192.168.5.7 und 192.168.5.12.

Diese Variable darf auch leer gelassen werden.

Wird diese Variable gleichzeitig mit `QOS_FILTER_x_IP_EXTERN` genutzt, so wird nur Verkehr gefiltert, der zwischen den in `QOS_FILTER_x_IP_INTERN` und `QOS_FILTER_x_IP_EXTERN` angegebenen IPs oder IP-Bereichen stattfindet.

Wichtig: *Falls zusätzlich durch `QOS_FILTER_x_OPTION` nach ACK, TOSMD, TOSMT, TOSMR oder TOSMC gefiltert wird und die Variable `QOS_CLASS_x_DIRECTION` der Zielklasse 'down' entspricht, dann wird diese Variable ignoriert.*

QOS_FILTER_x_IP_EXTERN Hier können IP-Adressen und IP-Bereiche aus dem externen Netzwerke (welches über `QOS_INTERNET_DEV` angebunden ist), nach denen gefiltert werden soll, angegeben werden. Sie sind durch Leerzeichen zu trennen und können frei kombiniert werden. Das ganze funktioniert analog zu `QOS_FILTER_x_IP_INTERN`.

Diese Variable darf auch leer gelassen werden.

Wichtig: *Falls zusätzlich durch `QOS_FILTER_x_OPTION` nach ACK, TOSMD, TOSMT, TOSMR oder TOSMC gefiltert wird und die Variable `QOS_CLASS_x_DIRECTION` der Zielklasse 'down' entspricht, dann wird diese Variable ignoriert.*

QOS_FILTER_x_PORT Hier können Ports und Portranges angegeben werden, getrennt durch Leerzeichen und dürfen frei kombiniert werden. Falls die Variable leer ist, werden Übertragungen über sämtliche Ports limitiert.

Zu dem Format von Portranges: Möchte man nach Ports von 5000 bis 5099 filtern, so würde das folgender Maßen aussehen:

`QOS_FILTER_x_PORT='5000-5099'`

Ein weiteres Beispiel: Man möchte den Verkehr über die Ports 20 bis 21, 137 bis 139 und Port 80 filtern und in die selbe Klasse stecken lassen. Das sähe dann so aus:

```
QOS_FILTER_x_PORT='20-21 137-139 80'
```

Diese Variable darf auch leer gelassen werden.

Wichtig:

- Wenn nach Ports gefilter wird, muß auch `QOS_FILTER_x_PORT_TYPE` etsprechend gesetzt werden.
- Wenn zusätzlich durch `QOS_FILTER_x_OPTION` nach ACK, TOSMD, TOSMT, TOSMR oder TOSMC gefiltert wird, dann werden Portranges ignoriert.

QOS_FILTER_x_PORT_TYPE Das setzen dieser Variable ist nur wichtig im Zusammenhang mit `QOS_FILTER_x_PORT` und muß auch nur dann gesetzt werden (wird ansonsten ignoriert).

Da sich die Ports beim Clientbetrieb von den Ports beim Serverbetrieb unterscheiden, muss angegeben werden ob der Port des Servers oder Clients gemeint ist. Als Bezugspunkte gelten hier die Rechner aus dem eigenen Netz.

Folgende Einstellungen sind Möglich:

```
QOS_FILTER_x_PORT_TYPE='client'
QOS_FILTER_x_PORT_TYPE='server'
```

Seit Version 2.1 ist auch die Kombination der beiden Argumenten möglich, um sowohl den Verkehr über den angegebenen Port aus dem eigenen Netz, als auch den Verkehr über selbigen Port aus dem Internet in die selbe Klasse zu stecken:

```
QOS_FILTER_x_PORT_TYPE='client server'
```

Dies entspricht der Erstellung von zwei ähnlichen Filtern, bei denen `QOS_FILTER_x_PORT_TYPE` einmal auf Client und einmal auf Server gesetzt wurde.

QOS_FILTER_x_OPTION Mit dieser Variable kann man weitere Eigenschaften für den Filter aktiveren. Es darf hier höchstens eines der folgenden Argumente angegeben werden, denn eine Kombination dieser in ein und dem selben Filter macht keinen Sinn. Hingegen ist es sehr wohl möglich und auch teilweise sinnvoll, daß zum Beispiel ein Filter für ACK-Pakete und ein 2. Filter für TOSMD-Pakete ihre Pakete in die selbe Zielklasse leiten (siehe `QOS_FILTER_x_CLASS`).

'ACK' Acknowledgement-Pakete.

Ein Paket das auf diese Option zutrifft, wird als Bestätigung für Datenpaket zurückgesendet. Wenn ihr z.B. einen großen Download am laufen habt, dann kommen bei euch viele Datenpakete an und für jedes muß eine Antwort gesendet werden, daß das Datenpaket angekommen ist. Lassen diese Bestätigungspakete auf sich warten, so so wartet der Datenversender ab, bis diese eingetroffen sind, bevor er neue Datenpakete sendet, was euch nicht so richtig schmeckt.

Das ganze ist insbesondere wichtig bei asymetrischen Verbindungen (ungleiche Up/Downstream-Bandbreite), wie sie bei den meisten privaten DSL-Angeboten üblich sind. Wird der meist relativ kleine Upstream an seine Grenzen gefahren, so stapeln sich die Pakete vor dem Ausgang förmlich auf und irgendwo in diesem riesigen Haufen sitzen hier und da die kleinen Bestätigungspakete. Im Normalfall geht

das dann hübsch der Reihe nach. Bis das Bestätigungspaket dann an der Reihe ist, kann es gut sein, daß es so lange gedauert hat, daß unser Datenversender eine kleine Pause einlegt und was wie gesagt nicht gut für den Downstream ist.

Wir müssen also dafür sorgen, daß die Bestätigungspakete auf die Überholspur kommen, so daß sie in windeseile an allen "normalen" Paketen vorbeihuschen, damit sie auch noch rechtzeitig beim Datenversender ankommen. Wie sich dies Option sinnvoll mit einer Klasse kombinieren läßt, wird bei den Anwendungsbeispielen erläutert.

ICMP Ping-Pakete (Protokoll ICMP)

Ping-Pakete werden dazu benutzt, die Zeit zu messen, die ein Paket von A nach B braucht. Wenn ihr also ordentlich angeben wollt, dann gebt den Ping-Paketen z.B. eine höhere Priorität. Das hat jetzt nichts mit dem Spielen im Internet selber zu tun, also nicht denken nur weil ihr den Ping-Pakete den Vorrang gibt, daß ihr super niedrige Pingzeiten im Spiel bekommt...

TCP TCP-Pakete (Protokoll TCP)

Es wird nur nach Paketen gefiltert, die das Protokoll TCP benutzen.

UDP UDP-Pakete (Protokoll UDP)

Es wird nur nach Paketen gefiltert, die das Protokoll UDP benutzen.

TOS* Type of Service

TOS steht für "Type of Service". Eine Applikation kann für jedes Paket was es verschickt eines der 4 TOS-Bits setzen. Damit wird angegeben welche Behandlung für die Pakete vorgesehen sind. So kann z.B. SSH TOS-Minimum-Delay für das Versenden der ein und Ausgabe setzen und TOS-Maximum-Troughput für das Versenden von Dateien. Generell benutzen Linux/Unix Programme diese Bits öfter als Windowsprogramme. Außerdem kann man z.B. auch it der Firewall die TOS-Bits für bestimmte Pakete setzen. Letztendlich kommt es dann aber auf die Router auf der Strecke an, ob die TOS-Bits beachtet werden, oder nicht. Wirklich von Interesse für einen fli4l sind aber eigentlich nur die TOS-Bits Minimum-Delay und Maximum-Throughput.

TOSMD - TOS Minimum-Delay Wird für Dienste benutzt, bei denen es wichtig ist, daß Pakete möglichst ohne Zeitverzögerung weitergeleitet werden. Empfolen wird dieses TOS-Bit für FTP (Kontrolldaten), Telnet und SSH.

TOSMT - TOS Maximum-Troughput Wird für Dienste benutzt, bei denen es wichtig ist, daß große Datenmengen mit hoher Geschwindigkeit weitergeleitet werden. Empfolen wird dieses TOS-Bit für FTP-Data und WWW.

TOSMR - TOS Maximum-Reliability Wird benutzt, wenn es wichtig ist, das man eine gewisse Sicherheit hat, daß die Daten an ihr Ziel gelangen, ohne das ein erneutes senden nötig ist. Empfolen wird dieses TOS-Bit für SNMP und DNS.

TOSMC - TOS Minimum-Cost Wird benutzt, wenn es wichtig ist die Kosten der Datenübertragung zu Minimieren. Empfolen wird dieses TOS-Bit für NNTP und SMTP.

4.19.2. Anwendungsbeispiele

Wie konfiguriert man OPT_QoS nun genau? Dies wird nun an einigen Beispielen gezeigt:

- Beispiel 1: Ein einfaches Beispiel mit dem Ziel die Bandbreite auf 3 Rechner zu verteilen.
- Beispiel 2: Ein Beispiel mit dem Ziel die Bandbreite auf 2 Rechner zu verteilen und die jeweiligen Bandbreiten pro Rechner wiederum noch ein zweites mal aufzuteilen auf einen Port und den restlichen Verkehr des jeweiligen Rechners.
- Beispiel 3: Ein Beispiel, welches die allgemeine Funktionsweise von QoS versucht nahezubringen.
- Beispiel 4: Beispielkonfiguration für das Bevorteilen von ACK-Paketen, damit der Downstream bei gleichzeitig starkem Upstream nicht einbricht.

Beispiel 1

Ein einfaches Beispiel mit dem Ziel die Bandbreite auf 3 Rechner zu verteilen.

Dazu erstellen wir 4 Klassen (siehe QOS_CLASS_N und folgende) mit den jeweiligen Geschwindigkeiten (siehe QOS_CLASS_x_MINBANDWIDTH / QOS_CLASS_x_MINBANDWIDTH) und haengen sie an die Klasse 0 (siehe QOS_CLASS_x_PARENT) also direkt an das Interface fuer “up” bzw. “down” (siehe QOS_CLASS_x_DIRECTION).

Die viert Klasse ist nur für eventuelle Besucher und bekommt weniger Bandbreite zugeteilt. Mit QOS_INTERNET_DEFAULT_DOWN=’4’ lassen wir in diesem Fall allen nicht gefilterten Verkehr in die vierte “Gast”-Klasse wandern. Da wir aber selten Gäste haben und die Bandbreite für die anderen 3 Klassen jeweils die selbe beträgt, bekommt jeder Rechner 1/3 der gesamten Bandbreite, effektiv also 256Kibit/s.

Mit dieser Konfiguration haben wir allerdings erst das Grundgeruest erstellt. Jetzt muessen wir noch sagen welcher Verkehr durch welche Klasse geregelt werden soll.

Dazu benutzen wir Filter, welche den Verkehr den einzelnen Klassen zuordnen. Wir erstellen also 3 Filter fuer die 3 Rechner (siehe QOS_FILTER_N und folgende) und Ordnen jeden Filter einer Klasse zu (siehe QOS_FILTER_x_CLASS). Jetzt koennen wir mit QOS_FILTER_x_IP_INTERN, QOS_FILTER_x_IP_INTERN, QOS_FILTER_x_PORT, QOS_FILTER_x_PORT und QOS_FILTER_x_OPTION bestimmen was durch die jeweilige Klasse zu der der Filter gehoert geregelt werden soll.

Nennen wir das Interface 0 und die 3 Klassen 1, 2 und 3 und die 3 Filter F1, F2 und F3 ergibt sich das in Abbildung 4.5 dargestellte Szenario.

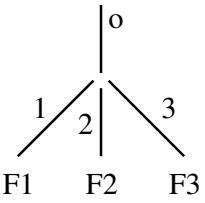


Abbildung 4.5.: Beispiel 1

Die Konfiguration sieht dann so aus:

Drei Rechner nach IP gefiltert die je 1/3 bekommen falls kein Gast anwesend ist:

```
OPT_QOS='yes'
QOS_INTERNET_DEV_N='1'
QOS_INTERNET_DEV_1='ppp0'
QOS_INTERNET_BAND_DOWN='768Kibit/s'
QOS_INTERNET_BAND_UP='128Kibit/s'
QOS_INTERNET_DEFAULT_DOWN='4'
QOS_INTERNET_DEFAULT_UP='0'

QOS_CLASS_N='4'

QOS_CLASS_1_PARENT='0'
QOS_CLASS_1_MINBANDWIDTH='232Kibit/s'
QOS_CLASS_1_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_1_DIRECTION='down'
QOS_CLASS_1_PRIO=''

QOS_CLASS_2_PARENT='0'
QOS_CLASS_2_MINBANDWIDTH='232Kibit/s'
QOS_CLASS_2_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_2_DIRECTION='down'
QOS_CLASS_2_PRIO=''

QOS_CLASS_3_PARENT='0'
QOS_CLASS_3_MINBANDWIDTH='232Kibit/s'
QOS_CLASS_3_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_3_DIRECTION='down'
QOS_CLASS_3_PRIO=''

QOS_CLASS_4_PARENT='0'
QOS_CLASS_4_MINBANDWIDTH='72Kibit/s'
QOS_CLASS_4_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_4_DIRECTION='down'
QOS_CLASS_4_PRIO=''


QOS_FILTER_N='3'

QOS_FILTER_1_CLASS='1'
QOS_FILTER_1_IP_INTERN='192.168.0.2'
QOS_FILTER_1_IP_EXTERN=''
QOS_FILTER_1_PORT=''
QOS_FILTER_1_PORT_TYPE=''
QOS_FILTER_1_OPTION=''

QOS_FILTER_2_CLASS='2'
QOS_FILTER_2_IP_INTERN='192.168.0.3'
QOS_FILTER_2_IP_EXTERN=''
QOS_FILTER_2_PORT=''
QOS_FILTER_2_PORT_TYPE=''
QOS_FILTER_2_OPTION=''

QOS_FILTER_3_CLASS='3'
QOS_FILTER_3_IP_INTERN='192.168.0.4'
```

```
QOS_FILTER_3_IP_EXTERN=''
QOS_FILTER_3_PORT=''
QOS_FILTER_3_PORT_TYPE=''
QOS_FILTER_3_OPTION=''
```

Die Option `QOS_INTERNET_DEFAULT_UP` wurde auf 0 gesetzt da der Upstream nicht beschraenkt werden soll.

Beispiel 2

Ein Beispiel mit dem Ziel die Bandbreite auf 2 Rechner zu verteilen und die jeweiligen Bandbreiten pro Rechner wiederum noch ein zweites mal aufzuteilen auf einen Port und den restlichen Verkehr des jeweiligen Rechners.

Dazu erstellen wir erst einmal wieder 2 Klassen mit den jeweiligen Gesamtgeschwindigkeit und haengen sie direkt an das Interface fuer "up" bzw. "down" (siehe erstes Beispiel). Jetzt erstellen wir fuer den ersten Rechner an der ersten Klasse zwei weitere Klassen. Die Klassen werden genau so erstellt wie die beiden ersten Klassen direkt am Interface, allerdings mit einer Besonderheit: `QOS_CLASS_x_PARENT` ist jetzt nicht 0, sondern die Nummer der Klasse an die die Klassen angehaengt werden sollen. Ist dies z. B. `QOS_CLASS_1`, so muss man jetzt `QOS_CLASS_1` von der Klasse die angehaengt werden soll auf 1 setzen. Das gleiche wird fuer den zweiten Rechner auch gemacht. Man haengt wieder zwei Klassen an die Klasse fuer den zweiten Rechner. Dies kann man nun nicht nur fuer zwei Rechner machen, sondern fuer so viele wie man moechte. Auch kann man so viele Unterklassen an einer Klasse erstellen wie man moechte.

Hiermit haben wir wieder das Grundgeruest erstellt und muessen nun mit den Filtern den Verkehr den einzelnen Klassen zuordnen. (siehe erstes Beispiel)

Wir erstellen also 2 Filter fuer den ersten Rechner und 2 Filter fuer den zweiten Rechner. Jeweils einen Filter fuer den Port und einen Filter fuer den restlichen Verkehr vom Rechner. Hierbei ist unbedingt auf die Reihenfolge zu achten. Als erstes jeweils nur den Port und danach den Rest. Anders herum wuerde ja schon der Filter fuer den Rest alles einer Klasse zuordnen.

Nennen wir das Interface 0 und die 6 Klassen 1, 2, 3, 4, 5 und 6 und die 4 Filter F1, F2, F3 und F4 ergibt sich das in Abbildung 4.5 dargestellte Szenario.

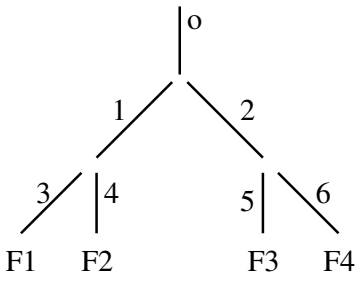


Abbildung 4.6.: Beispiel 2

Die Konfiguration sieht dann so aus:

Zwei Klassen fuer 2 Rechner die je 1/2 bekommen, und zwar vom Interface, mit jeweils 2 Klassen fuer einen Port der 2/3 bekommt und den Rest der 1/3 bekommt, und zwar jeweils von der Vaterklasse:

```
OPT_QOS='yes'
```

```
QOS_INTERNET_DEV_N='1'
QOS_INTERNET_DEV_1='ppp0'
QOS_INTERNET_BAND_DOWN='768Kibit/s'
QOS_INTERNET_BAND_UP='128Kibit/s'
QOS_INTERNET_DEFAULT_DOWN='7'
QOS_INTERNET_DEFAULT_UP='0'

QOS_CLASS_N='6'

QOS_CLASS_1_PARENT='0'
QOS_CLASS_1_MINBANDWIDTH='384Kibit/s'
QOS_CLASS_1_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_1_DIRECTION='down'
QOS_CLASS_1_PRIO=''

QOS_CLASS_2_PARENT='0'
QOS_CLASS_2_MINBANDWIDTH='384Kibit/s'
QOS_CLASS_2_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_2_DIRECTION='down'
QOS_CLASS_2_PRIO=''

QOS_CLASS_3_PARENT='1'
QOS_CLASS_3_MINBANDWIDTH='256Kibit/s'
QOS_CLASS_3_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_3_DIRECTION='down'
QOS_CLASS_3_PRIO=''

QOS_CLASS_4_PARENT='1'
QOS_CLASS_4_MINBANDWIDTH='128Kibit/s'
QOS_CLASS_4_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_4_DIRECTION='down'
QOS_CLASS_4_PRIO=''

QOS_CLASS_5_PARENT='2'
QOS_CLASS_5_MINBANDWIDTH='256Kibit/s'
QOS_CLASS_5_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_5_DIRECTION='down'
QOS_CLASS_5_PRIO=''

QOS_CLASS_6_PARENT='2'
QOS_CLASS_6_MINBANDWIDTH='128Kibit/s'
QOS_CLASS_6_MAXBANDWIDTH='768Kibit/s'
QOS_CLASS_6_DIRECTION='down'
QOS_CLASS_6_PRIO=''

QOS_FILTER_N='4'

QOS_FILTER_1_CLASS='3'
QOS_FILTER_1_IP_INTERN='192.168.0.2'
QOS_FILTER_1_IP_EXTERN=''
QOS_FILTER_1_PORT='80'
QOS_FILTER_1_PORT_TYPE='client'
QOS_FILTER_1_OPTION=''
```

```
QOS_FILTER_2_CLASS='4'
QOS_FILTER_2_IP_INTERN='192.168.0.2'
QOS_FILTER_2_IP_EXTERN=''
QOS_FILTER_2_PORT=''
QOS_FILTER_2_PORT_TYPE=''
QOS_FILTER_2_OPTION=''

QOS_FILTER_3_CLASS='5'
QOS_FILTER_3_IP_INTERN='192.168.0.3'
QOS_FILTER_3_IP_EXTERN=''
QOS_FILTER_3_PORT='80'
QOS_FILTER_3_PORT_TYPE='client'
QOS_FILTER_3_OPTION=''

QOS_FILTER_4_CLASS='6'
QOS_FILTER_4_IP_INTERN='192.168.0.3'
QOS_FILTER_4_IP_EXTERN=''
QOS_FILTER_4_PORT=''
QOS_FILTER_4_PORT_TYPE=''
QOS_FILTER_4_OPTION=''
```

Bei diesem Beispiel wurde die Option `QOS_INTERNET_DEFAULT_DOWN` so gewaehlt, daß der Verkehr, welcher nicht durch einen Filter einer Klasse zugeordnet wird, in eine nicht existierende Klasse gesteckt wird. Einfach aus dem Grund um das Beispiel zu vereinfachen und weil in dem Beispiel davon ausgegangen wird dass es keinen Rest gibt. Verkehr der in eine nicht existierende Klasse geleitet wird, wird nur sehr langsam weiter geleitet. Wenn es einen Rest gibt, ist also unbedingt darauf zu achten, daß dieser in eine eigene Klasse gesteckt wird, die auch existiert.

Die Option `QOS_INTERNET_DEFAULT_UP` wurde auf 0 gesetzt da der Upstream nicht beschraenkt werden soll.

Beispiel 3

Ein Beispiel, welches die allgemeine Funktionsweise von QoS versucht nahezubringen.

In Abbildung 4.7 ist noch einmal die Aufteilung aus dem zweiten Beispiel zu sehen, allerdings mit einer Erweiterung. An die Beiden Unterklassen der zweiten Klasse sind jeweils noch zwei weitere Unterklassen Angehaengt. Es ist also moeglich noch tiefer zu verschachteln. Es ist moeglich, tiefer zu verschachteln als auf diesem Bild, die momentane Grenze liegt hier bei 8 Stufen, also man darf maximal nur 7 Weitere Stufen nach dem Interface erstellen, danach ist schluss. In die “Breite” ist allerdings keine Grenze gesetzt. Man kann also an eine Unterklasse innerhalb einer Stufe so viele Klassen anhaengen wie man moechte.

Auf diesem Bild ist ausserdem noch zu erkennen, dass es auch moeglich ist mehr als einen Filter an eine Klasse zu haengen, so wie es an der Klasse 10 geschehen ist. Aber auch bei den Filtern bleibt zu beachten dass es Momentan nicht moeglich ist einen Filter mitten in den “Baum” zu heangen, so wie es mit F8 geschehen sollte.

Schauen wir uns nun als letztes noch den Sinn von Klassen und Unterklassen an. Klassen dienen dazu die Geschwindigkeit einer Verbindung einzustellen und zu regeln. Die Verteilung der Geschwindigkeit erfolgt wie bei `QOS_CLASS_x_MINBANDWIDTH` beschrieben. Allerdings kann dies einen Nachteil haben wenn man z.b. alle Klassen an eine Klasse haengt. Moechte man z.b.

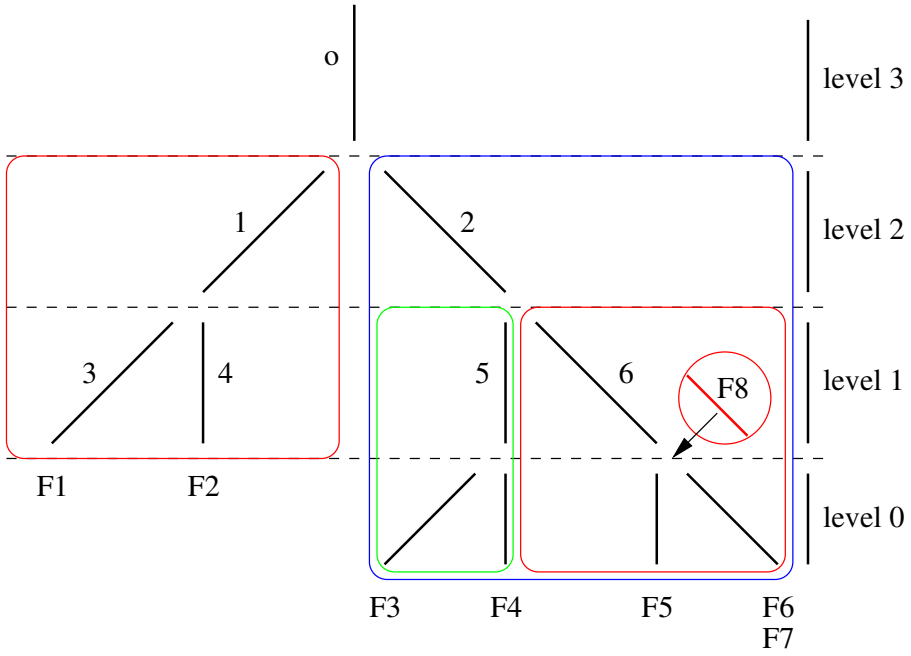


Abbildung 4.7.: Beispiel 3

einem Rechner die haelfte der bandbreite geben und dem zweiten ebenfalls die haelfte allerdings aufgeteilt auf 2/3 http und 1/3 Rest also jeweils 2/6 und 1/6 vom ganzen. So geschieht nun folgendes: bei Vollast bekommt jeder seine haelfte. Uebertraegt der zweite jedoch nichts ueber http so bleibt ja 2/6 ueber. Diese 2/6 bekommt jedoch nun nicht der 2. Rechner alleine, sondern es wird nach dem beschriebenen Verfahren aufgeteilt. Um dieses zu verhindern erstellt man Unterklassen. Der Verkehr einer Klasse wird somit erst an die Unklassen verteilt, erst wenn diese nicht den Kompletten Verkehr beanspruchen wird der Rest an andere Klassen Verteilt. In dem Bild sind jeweils die Bereiche eingekreist welche zusammengehoren. Rot = 1, Blau = 2, Gruen = 5 und Orange = 6.

Beispiel 4

Beispielkonfiguration für das Priorisieren von ACK-Paketen, damit der Downstream bei gleichzeitig starkem Upstream nicht einbricht:

```
OPT_QOS='yes'
QOS_INTERNET_DEV_N='1'
QOS_INTERNET_DEV_1='ppp0'
QOS_INTERNET_BAND_DOWN='768Kibit/s'
QOS_INTERNET_BAND_UP='128Kibit/s'
QOS_INTERNET_DEFAULT_DOWN='0'
QOS_INTERNET_DEFAULT_UP='2'
```

Hier konfigurieren wir ppp0 als Internetdevice (DSL) und geben die für TDSL (und einiger anderer Provieder) übliche Up/Downstreambandbreiten an. Eventuell ist es nötig, daß wir

die Upstreambandbreite noch um das eine oder andere Kibibit herabsetzen, das muß man ausprobieren.

Da wir keine Klassen für den Downstream einrichten wollen, setzen wir

```
QOS_INTERNET_DEFAULT_DOWN='0'
```

Für den Upstream soll die Klasse mit der Nummer 2 die Standardklasse sein. Das Netzwerkdevice ist eth0 und auf 10Mibit/s eingestellt.

```
 QOS_CLASS_N='2'

QOS_CLASS_1_PARENT='0'
QOS_CLASS_1_MINBANDWIDTH='127Kibit/s'
QOS_CLASS_1_MAXBANDWIDTH='128Kibit/s'
QOS_CLASS_1_DIRECTION='up'
QOS_CLASS_1_PRIO=''
```

Dies ist die Klasse in die wir unsere ACK-(Bestätigungs-)Pakete hineinstecken wollen. Die ACK-Pakete sind recht klein und benötigen deswegen nur recht wenig Bandbreite. Trotzdem wollen wir sie eigentlich in keinster Weise einschränken und teilen ihr 127Kibit/s zu. 1Kibit/s lassen wir übrig für den Rest.

```
QOS_CLASS_2_PARENT='0'
QOS_CLASS_2_MINBANDWIDTH='1Kibit/s'
QOS_CLASS_2_MAXBANDWIDTH='128Kibit/s'
QOS_CLASS_2_DIRECTION='up'
QOS_CLASS_2_PRIO=''
```

In diese Klasse soll dann der Rest (alles außer ACK-Pakete) hineingesteckt werden. Die Bandbreite, die wir dieser Klasse zuteilen sind die verbleibenden 1Kibit/s (128-127=1). Wir begrenzen sie aber auch nicht auf 1Kibit/s, begrenzt wird die Klasse durch den Eintrag

```
QOS_CLASS_2_MAXBANDWIDTH='128Kibit/s'
```

Da unsere erste Klasse die zugeteilte Bandbreite wohl kaum ausnutzen wird, bleibt also immer etwas übrig und das was übrig bleibt schnappt sich dann die zweite. Wenn man den Upstream noch weiter aufteilen möchte (was meist der Fall ist), so sind alle weiteren Klassen unter diese Klasse zu "hängen". Dabei muß natürlich auch `QOS_INTERNET_DEFAULT_UP` entsprechend angepaßt werden.

```
QOS_FILTER_N='1'

QOS_FILTER_1_CLASS='1'
QOS_FILTER_1_IP_INTERN=''
QOS_FILTER_1_IP_EXTERN=''
QOS_FILTER_1_PORT=''
QOS_FILTER_1_PORT_TYPE=''
QOS_FILTER_1_OPTION='ACK'
```

Dieser Filter filtert alle Pakete, die auf die Option ACK zutreffen, also ACK-Pakete. Durch den Eintrag `QOS_FILTER_1_CLASS`='1' erreichen wir, daß diese gefilterten Pakete in die 1. Klasse gesteckt werden.

Zum Testen muß sucht man sich am besten eine gute oder mehrere Up- und Downloadquellen aus, von denen man weiß, daß sie sowohl den Up- als auch den Downstream voll auslasten können und läßt die Kabel glühen. Dabei sollte man einen Blick auf die Trafficanzeige des ImonC werfen. Am besten führt man das ganze auch mal ohne QoS durch.

Der Downstream sollte gar nicht oder wesentlich weniger stark einbrechen als ohne diese Konfiguration. Wie schon gesagt kann man die Lage noch verbessern, in dem man die Upstreambandbreite in Kibibit-Schritten herabsetzt und dann die Auswirkungen beobachtet. Bei mir wurde zum Beispiel das Optimum bei 121Kibit/s erreicht (kein Einbruch des Downstreams mehr). Dabei sind natürlich auch die MAXBANDWIDTH- und MINBANDWIDTH-Werte der Klassen entsprechend anzupassen.

4.20. SSHD - Secure Shell, Secure Copy

Eine Secure-Shell bietet die Möglichkeit, eine verschlüsselte Verbindung mit dem fli4l-Router aufzunehmen. Außerdem können mit dem Secure-Copy-Befehl Dateien verschlüsselt auf den fli4l-Router übertragen werden. Wird zusätzlich eine Public Key Anmeldung (Seite 236) benutzt , können Befehle auf dem fli4l-Router und Dateiübertragungen auch scriptgesteuert ausgeführt werden. Ab der Version 2.1.7 gibt es nur noch einen SSH2 Server.

4.20.1. Installation des Secure-Shell-Dienstes

OPT_SSHD Standard-Einstellung: `OPT_SSHD='no'`

Soll der Zugriff auf den Router mittels ssh ermöglicht werden, bedarf es der Änderung auf von `OPT_SSHD` auf `'yes'`. Dies installiert den ssh-Server Dropbear auf dem fli4l-Router.

SSHD_ALLOWPASSWORDLOGIN Standard-Einstellung: `SSHD_ALLOWPASSWORDLOGIN='yes'`

Wird `SSHD_ALLOWPASSWORDLOGIN` auf `'no'` eingestellt, ist die Anmeldung mit ssh über ein Password auf dem fli4l-Router nicht mehr möglich. Die Anmeldung kann dann nur noch mittels privatem/öffentlichem Schlüsselpaar (privat/public key) erfolgen. Dies setzt voraus, daß ein öffentlicher Schlüssel (Seite 236) auf dem Router hinterlegt ist.

SSHD_CREATEHOSTKEYS Standard-Einstellung: `SSHD_CREATEHOSTKEYS='no'`

Ein ssh-Server benötigt einen sogenannten Hostkey, der weltweit einmalig sein sollte, damit sich der ssh-Server eindeutig gegenüber einem ssh-Client identifizieren kann. Das sshd opt-Paket liefert zwar einen Hostkey mit, um das erste Einloggen auf dem fli4l-Router per ssh-Client zu erlauben, aber der mitgelieferte Hostkey sollte so schnell wie möglich durch einen selbst generierten, nur Ihnen bekannten Hostkey ersetzt werden. Die Generierung eines eigenen Hostkeys ist deshalb so wichtig, weil nur auf diese Weise Schutz gegen so genannte Man-in-the-Middle-Attacken möglich ist. Ihr ssh-Client bemerkt es, wenn ein Cracker vorgibt, Ihr fli4l-Router zu sein, da dem Cracker Ihr Hostkey nicht bekannt ist. Ihr ssh-Client warnt Sie daraufhin mit einer Meldung, dass der Hostkey sich geändert hat.

Die Erzeugung Ihres eigenen Hostkeys geschieht vollkommen automatisch, sobald Sie die Einstellung `SSHD_CREATEHOSTKEYS` auf `'yes'` setzen. Dieser Vorgang ist sehr rechenintensiv und kann deshalb die Bootzeit um mehrere Minuten verlängern. Wenn der fli4l-Router mit aktiviertem `SSHD_CREATEHOSTKEYS` Eintrag startet, wird ein (oder mehrere) Hostkey(s)

in dem Verzeichnis `/tmp/ssh` erzeugt. Die Dateien die dort stehen, kopieren Sie in das Verzeichnis `etc/ssh` unterhalb Ihres config Verzeichnisses. In meinem Fall sieht ein Directorylisting des config.babel Verzeichnisses so aus:

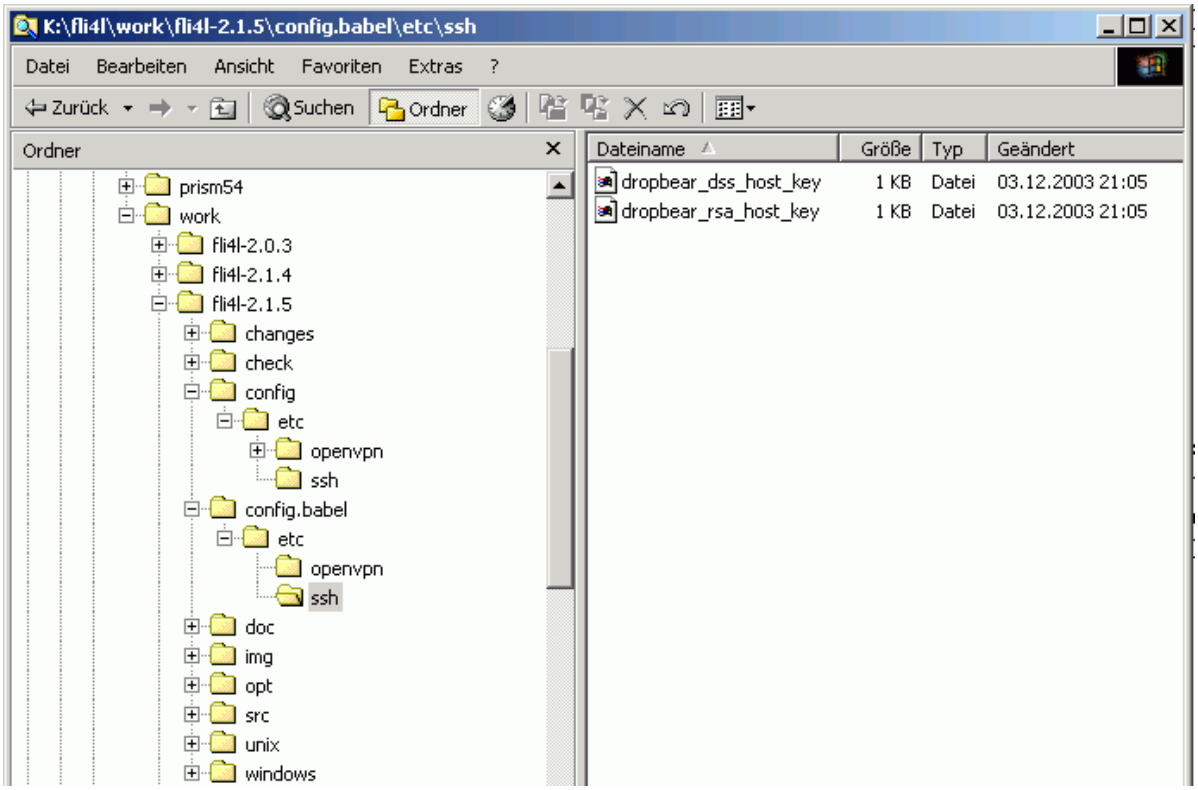


Abbildung 4.8.: Verzeichnisstruktur fli4l

Beachten Sie, dass unterhalb der `config.babel` Verzeichnis erst das Verzeichnis etc kommt und darunter dann das Verzeichnis ssh. Und genau dorthin muss der oder die eben erzeugte(n) Hostkey(s) kopiert werden. Ab der fli4l Version 2.1.5 werden Dateien die unterhalb Ihres config Verzeichnisses stehen vorangig vor den Dateien aus dem opt Verzeichnis behandelt. Dadurch werden bei dem nächsten Update Ihres fli4l-Routers die Dateien aus dem Verzeichnis `config/etc/ssh` eingebunden und nicht die Dateien, die im Verzeichnis `opt/etc/ssh` stehen. So ist es möglich für jeden fli4l-Router, den Sie konfigurieren, einen eigenen Hostkey zu benutzen. Wenn Sie die fli4l-Routerdateien erzeugen, erscheint ziemlich zum Schluss die Meldung „appending config specific files to opt.img ...“. Dort werden dann alle Dateien aufgelistet, die aus Ihrem config Verzeichnis kommen und nicht aus dem opt Verzeichnis.

```
#
# appending config specific files to opt.img ...
#
etc/ssh/dropbear_dss_host_key
etc/ssh/dropbear_rsa_host_key
```

Wenn Sie einen neuen Hostkey erzeugt haben, setzen Sie danach den Wert `SSHD_-`

CREATEHOSTKEYS wieder auf 'no', um Platz auf der fli4l Boot-Diskette zu sparen und damit die Startskripte des fli4l-Routers nicht ständig einen neuen Hostkey generieren.

Wenn Sie sich nach dem Update des Hostkey auf Ihrem fli4l-Router anmelden, wird eine (je nach Programm unterschiedliche) Warnmeldung von Ihrem ssh-Client ausgegeben, die Sie auf einen geänderten Hostkey hinweist. Das ist normal, da Sie ja gerade den von fli4l mitgelieferten Hostkey gegen den von Ihnen erzeugten Hostkey ausgetauscht haben. Befolgen Sie die Hinweise Ihres ssh-Client, wie Sie den geänderten Hostkey permanent übernehmen können. Sollten Sie diese Warnmeldung zu einem späteren Zeitpunkt noch einmal bekommen, sollten Sie in jedem Fall prüfen, warum diese Warnung ausgegeben wurde und nicht einfach blind den geänderten Hostkey akzeptieren.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
ca:a4:ab:e7:af:d8:68:05:d3:1f:e6:15:08:d6:ed:36.
Please contact your system administrator.
Add correct host key in /home/babel/.ssh/known_hosts to get rid of this message.
Offending key in /home/babel/.ssh/known_hosts:7
Password authentication is disabled to avoid man-in-the-middle attacks.
```

SSHD_PORT Standard-Einstellung: SSHD_PORT='22'

Mit SSHD_PORT kann abweichend vom Standard ein Port angegeben werden, auf dem der ssh-Server laufen soll.

Möchte man den ssh-Zugang auch von aussen erlauben, ist INPUT_ACCEPT_PORT_x (Seite 41) anzupassen.

Die Befehle, um von einem Unix-/Linux-Rechner über das SSH-Protokoll auf fli4l zuzugreifen, lauten:

- ssh - Secure Shell
- scp - Secure Copy

Entsprechende Programme für Windows sind ebenso verfügbar, s. auch:
http://www.chiark.greenend.org.uk/~sgtatham/putty/
http://winscp.net/eng/docs/lang:de
http://www.tectia.com/de/de.iw3

SSHD_PUBLIC_KEYS_N Standard-Einstellung: SSHD_PUBLIC_KEYS_N='0'

SSHD_PUBLIC_KEYS_N beschreibt die Anzahl der öffentlichen Schlüssel, die auf den fli4l-Router kopiert werden sollen.

SSH gestattet die Authentifizierung mit Hilfe von asymmetrischen Verschlüsselungsverfahren. Dabei erfolgt die Authentifizierung anstatt über Nutzername und Passwort über

Nutzername und einem Public-/Privatkey. Damit kann man sich die Eingabe eines Passwortes sparen. Das Schlüsselpaar erzeugt man mit Hilfe von ssh-keygen (oder puttygen, wenn putty unter Windows als ssh-Client eingesetzt wird). Optional kann beim Schlüsselerzeugen eine Passphrase vergeben (also ein Password, das man braucht, wenn man den Schlüssel benutzen will) werden, welches die Sicherheit noch zusätzlich erhöht. Benutzt man Passphrases sollte man über den Einsatz eines Schlüsselagenten nachdenken (siehe ssh-agent oder pageant).

Wichtig: *Der private Teil des Schlüsselpaares, ist so sorgfältig zu behandeln wie ein Passwort, da er die gleiche Funktion erfüllt. Der private Teil des Schlüssel wird bei dem ssh-Client hinterlegt. Der öffentliche Teil des Schlüssel wird für den fli4l-Router gebraucht und mit* `SSHD_PUBLIC_KEY_x` *oder* `SSHD_PUBLIC_KEYFILE_x` *zur Verfügung gestellt.*

Für weitere Informationen siehe die manual Pages von ssh und Konsorten bzw. die Dokumentation zu putty (http://www.chiark.greenend.org.uk/~sgtatham/putty/).

SSHD_PUBLIC_KEY_x Für jeden Nutzer, der über ssh Zugang zum fli4l-Router erlangen möchte, kann hier der öffentliche Teil des Schlüssel angegeben werden. Am einfachsten geht das per Cut-and-Paste aus einem Terminalfenster heraus. Das könnte z.B. in etwa wie folgt aussehen:

```
SSHD_PUBLIC_KEY_1='1024 ... nutzername@hostname'
```

Wichtig: *Der Schlüssel enthält keine Zeilenumbrüche. Bei Cut-and-Paste aus puttygen heraus werden aber eventuell selbige eingefügt. Diese Zeilenumbrüche müssen wieder entfernt werden.*

SSHD_PUBLIC_KEYFILES_N Standard-Einstellung: `SSHD_PUBLIC_KEYFILES_N='0'`

Anstatt den Inhalt des öffentlichen Teil des Schlüssel in die sshd.txt Datei zu kopieren, können Sie den öffentlichen Teil des Schlüssel auch direkt in das opt-Archiv kopieren lassen. Das funktioniert genauso wie bei `SSH_CREATEHOSTKEYS` beschrieben wurde. Kopieren Sie Ihren öffentlichen Teil des Schlüssel in das Verzeichnis `<config>/etc/ssh`.

SSHD_PUBLIC_KEYFILE_x Der Dateiname des öffentlichen Teil des Schlüssels im `<config>/etc/ssh` Verzeichnis.

```
SSHD_PUBLIC_KEYFILE_1='root@fli4l'
```

SSHD_PRIVATE_KEYFILES_N Standard-Einstellung: `SSHD_PRIVATE_KEYFILES_N='0'`

Wenn sie mit dem ssh oder plink Client private Schlüssel zur Anmeldung an einen ssh Server benutzen wollen können sie diese in das Verzeichnis config/etc/ssh kopieren. Das funktioniert genauso wie bei `SSH_CREATEHOSTKEYS` beschrieben wurde. Kopieren sie Ihren privaten Teil des Schlüssel in das Verzeichnis `<config>/etc/ssh`. Private Schlüssel im OpenSSH Format werden automatisch bei jedem Startvorgang von fli4l ins das dropbear Format konvertiert. Dieser Luxus kostet ca. 4 kB Platz im opt–Archiv. Wollen Sie diesen Platz einsparen, muss der Schlüssel im dropbear Format vorliegen.

SSHD_PRIVATE_KEYFILE_x Der Dateiname des privaten Teil des Schlüssels im <config>/etc/ssh Verzeichnis.

```
SSHD_PRIVATE_KEYFILE_1='babel@rootserver'
```

4.20.2. Installation des Secure-Copy-Dienstes

OPT_SCP Standard-Einstellung: `OPT_SCP='no'`

Wie oben bereits erwähnt, kann man mit Hilfe von ssh auch Dateien mit einem ssh-Client auf den Router kopieren. Möchte man dies tun, muss man `OPT_SCP` auf `'yes'` setzen. Um Dateien über SCP vom Router auf einen anderen Computer kopieren zu können, ist es notwendig zusätzlich `OPT_SSH_CLIENT` auf `'yes'` zu setzen.

Der dbclient kennt einige Parameter eines kompletten ssh Clients nicht. Daher gibt es folgende Warnungen wenn man scp benutzt. Die können aber ignoriert werden. In einer späteren Version des dbclient sollen diese Warnhinweise laut Author vom dbclient entfernt werden!

```
WARNING: Ignoring unknown argument '-x' WARNING: Ignoring unknown
argument '-oForwardAgent no' WARNING: Ignoring unknown argument
'-oClearAllForwardings yes'
```

4.20.3. Installation des dbclients

OPT_SSH_CLIENT Standard-Einstellung: `OPT_SSH_CLIENT='no'`

Wenn man einen reinen ssh2 Client benutzen möchte, kann man den dbclient von dropbear durch setzen von `OPT_SSH_CLIENT='yes'` aktivieren. Dieser Client hat den Vorteil, dass er sich viel Programmcode mit dem dropbear ssh Server teilt. Dadurch wird sehr viel Platz im opt–Archiv gespart. Der dbclient ist weitgehend kompatibel mit dem ssh Client, die Befehlsparameter sind ähnlich. Es wird auch ein symbolischer Link auf /usr/bin/ssh angelegt damit ein gewohntes ssh <host> funktioniert.

Wenn man die dblcient bekannten Hostkeys permanent speichern will muss man die Datei known_hosts auf dem Verzeichnis /.ssh auf dem fli4l–Router in das config/etc/ssh kopieren. Das geschieht ähnlich wie mit einem erzeugten Hostkey. In dem folgenden Beispiel ist das ausgepackte fli4l Verzeichnis (in der die fli4l-Bootdiskette erzeugt wird) in /home/babel/fli4l-3.6.2 zu finden. Die Konfigurationsdateien liegen alle im Verzeichnis config.babel.

```
cd /home/babel/fli4l-3.6.2
mkdir -p config.babel/etc/ssh
scp fli4l:/.ssh/* config.babel/etc/ssh
```

4.20.4. Installation des plink Clients

OPT_PLINK_CLIENT Standard-Einstellung: `OPT_PLINK_CLIENT='no'`

Installiert auf dem fli4l-Router einen ssh1/ssh2/telnet Client. Das plink Programm ist die Unixversion des bekannten PuTTY Programms für Windows. Ein Aufruf von plink auf dem fli4l-Router gibt eine Hilfeseite für die Benutzung von plink aus.

Wenn man die plink bekannten Hostkeys permanent speichern will muss man die Datei sshhostkeys auf dem Verzeichnis /.putty auf dem fli4l–Router in das config/etc/plink kopieren. Das geschieht ähnlich wie mit einem erzeugten Hostkey. In dem folgenden Beispiel ist das ausgepackte fli4l Verzeichnis (in der die fli4l-Bootdiskette erzeugt wird) in /home/babel/fli4l-3.6.2 zu finden. Die Konfigurationsdateien liegen alle im Verzeichnis config.babel.

```
cd /home/babel/fli4l-3.6.2
mkdir -p config.babel/etc/plink
scp fli4l:/.putty/* config.babel/etc/plink
```

4.20.5. Installation des sftp-server

OPT_SFTPSERVER Standard-Einstellung: `OPT_SFTPSERVER='no'`

Installiert auf dem fli4l-Router einen sftp-server.

4.20.6. Literatur

Dropbear SSH2 Site: http://matt.ucc.asn.au/dropbear/dropbear.html
Claas Hilbrecht <babel@fli4l.de>, im April 2004

4.21. TOOLS - Zusätzliche Werkzeuge zum Debugging

Das Paket TOOLS liefert eine Reihe von Unix Programmen, die zumeist für Administrations- und Debugzwecke gedacht sind. Andere Programme wie wget werden z.B. dafür verwendet, die erste (Werbungs-)Seite einiger Provider abzufangen. Mit dem Wert 'yes' wird das jeweilige Programm mit auf den fli4l-Router kopiert. Die Standardeinstellung ist 'no'. Die Programme werden nur kurz vorgestellt, wie sie zu bedienen sind, entnehme man bitte den man Pages einer beliebigen Unix/ Linux Distribution oder online unter: http://www.linuxmanpages.com

4.21.1. Die Hardware-Erkennung (Experimentell)

Oftmals weiß man nicht genau, welche Hardware im eigenen Rechner steckt bzw. welche Treiber man nun genau für seine Netzwerkkarte oder seinen USB-Chipsatz verwenden soll. Die Hardware kann an der Stelle helfen. Sie liefert eine Liste von Geräten im Rechner und wenn möglich den dazugehörenden Treiber. Man kann dabei auswählen, ob die Erkennung gleich beim Booten erfolgen soll (was sich vor einer Erstinstallation empfiehlt) oder später bei laufendem Rechner bequem über das Web-Interfaces getriggert werden soll. Die Ausgabe könnte dabei z.B. wie folgt aussehen:

```
fli4l 3.6.2 # cat /bootmsg.txt
#
# PCI Devices and drivers
#
Host bridge: Advanced Micro Devices [AMD] CS5536 [Geode companion] Host Bridge (rev 33)
Driver: 'unknown'
Entertainment encryption device: Advanced Micro Devices [AMD] Geode LX AES Security Block
Driver: 'geode_rng'
Ethernet controller: VIA Technologies, Inc. VT6105M [Rhine-III] (rev 96)
```

```
Driver: 'via_rhine'
Ethernet controller: VIA Technologies, Inc. VT6105M [Rhine-III] (rev 96)
Driver: 'via_rhine'
Ethernet controller: VIA Technologies, Inc. VT6105M [Rhine-III] (rev 96)
Driver: 'via_rhine'
Ethernet controller: Atheros Communications, Inc. AR5413 802.11abg NIC (rev 01)
Driver: 'unknown'
ISA bridge: Advanced Micro Devices [AMD] CS5536 [Geode companion] ISA (rev 03)
Driver: 'unknown'
IDE interface: Advanced Micro Devices [AMD] CS5536 [Geode companion] IDE (rev 01)
Driver: 'amd74xx'
USB Controller: Advanced Micro Devices [AMD] CS5536 [Geode companion] OHC (rev 02)
Driver: 'ohci_hcd'
USB Controller: Advanced Micro Devices [AMD] CS5536 [Geode companion] EHC (rev 02)
Driver: 'ehci_hcd'
```

Hier stecken also im wesentlichen 3 Netzwerkkarten drin, die vom 'via_rhine'-Treiber verwaltet werden und eine Atheros-Wlan-Karte, die vom madwifi-Treiber verwaltet wird (der Name wird noch nicht korrekt aufgelöst).

OPT_HW_DETECT Diese Variable sorgt dafür, daß die für die Hardware-Erkennung Dateien auf dem Router landen. Man kann sich die Ergebnisse dann entweder nach dem Booten auf der Konsole ansehen, wenn man `HW_DETECT_AT_BOOTTIME` auf 'yes' gesetzt hat oder im Web-Interface ansehen, wenn man `OPT_HTTPD` (Seite 131) auf 'yes' gesetzt hat. Im Web-Interface kann man sich natürlich auch den Inhalt von '/bootmsg.txt' ansehen, wenn man schon einen funktionierenden Netzzugang hat.

HW_DETECT_AT_BOOTTIME Startet die Hardware-Erkennung beim Booten. Die Erkennung läuft im Hintergrund (sie dauert ein wenig) und schreibt dann ihre Ergebnisse auf die Konsole und nach '/bootmsg.txt'.

4.21.2. Die Tools

OPT_ARP Ansehen und Manipulieren der ARP Tabelle

Den Befehl arp verwendet man, um die Ethernetadresstabelle des Kernels zu bearbeiten oder anzusehen.

OPT_BCRELAY Weiterleiten von Broadcasts

Mit dem Programm bcrelay können Broadcasts zwischen Interfaces weitergeleitet werden. Beispiel:

```
bcrelay -d -i eth0 -o eth1
```

OPT_E3 Ein Editor für fli4l

Dies ist ein sehr kleiner, in Assembler geschriebener Editor. Er stellt verschiedene Editor-Modi zur Verfügung, die andere („große") Editoren nachstellen. Um einen bestimmten Modus zu wählen, reicht es e3 mit dem richtigen Befehl zu starten. Eine Kurzübersicht der Tastenbelegung bekommt man, wenn man e3 ohne Parameter startet oder Alt+H drückt

(außer im VI-Modus, dort muß man im CMD-Modus „:h” eintippen). Zu beachten ist auch, daß das Caret-Zeichen (^) für die Ctrl-/Strg-Taste steht.

Befehl	Modus
e3 / e3ws	WordStar, JOE
e3vi	VI, VIM
e3em	Emacs
e3pi	Pico
e3ne	NEdit

OPT_FTP FTP-Client

OPT_IFTOP Netzwerküberwachung

Mit dem Programm iftop wird eine Auflistung alle aktiven Netzwerkverbindung und deren Durchsatz direkt auf dem fli4l angezeigt.

Das Programm iftop wird nach dem Anmelden auf dem fli4l-Router durch Eingabe von iftop gestartet.

OPT_IMONC Textorientiertes Steuerprogramm für imond

Dieses Programm liefert ein textorientiertes Frontend für den Router, um den imond zu steuern.

OPT_IPERF Performancemessung im Netzwerk

Mit dem Programm iperf kann eine Performancemessung des Netzwerks durchgeführt werden. Dazu wird das Programm auf den beiden beteiligten Testsystemen gestartet. Auf dem Server wird das Programm mit

```
fli4l-server 3.6.2~# iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
```

gestartet. Der Server wartet dann auf eine Verbindung vom Client. Der Client wird durch

```
fli4l-client 3.6.2~# iperf -c 1.2.3.4
------------------------------------------------------------
Client connecting to 1.2.3.4, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 1.2.3.5 port 50311 connected with 1.2.3.4 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec    985 MBytes    826 Mbits/sec
```

gestartet. Sofort startet die Performancemessung und zeigt die ersten Ergebnisse an. iperf kennt noch eine Reihe weiterer Optionen, für Details schauen Sie sich bitte die Informationen auf der Homepage http://iperf.sourceforge.net/ an.

Hinweis: Die aktuelle Version von iperf unterstützt keine bidirektionale Messung, weil es Probleme mit der Threadlibrary der uClibc gab.

OPT_LSPCI Auflisten aller PCI-Geräte

OPT_MTOOLS Die mtools stellen eine Reihe von DOS-ähnlichen Befehlen zum vereinfachten Umgang (Kopieren, Formatieren, etc.) mit Disketten bereit. Zusätzlich sind noch einige Programme aus den fdutils (z.B. superformat) mit aufgenommen. Das Shellscript `fdformat.sh` formatiert eine Diskette im 1680er Format. Die Anwendung des Scripts erfordert keine weiteren Kenntnisse der Einzelbefehle.

Die genaue Syntax der Befehle kann in der Dokumentation von mtools bzw. fdutils nachgeschlagen werden:
http://www.gnu.org/software/mtools/manual/mtools.html
http://fdutils.linux.lu/Fdutils.html

OPT_NETCAT Übertragen von Daten an TCP basierte Server

OPT_NETSTAT Auflisten von Netzwerkinformationen

netstat --inet zum Beispiel liefert eine auführliche Liste aller internetbasierten Verbindungen auf dem Rechner. Damit lassen sich auch Verbindungen darstellen, die zwischen lokal laufenden Anwendungen bestehen.

OPT_NTTCP Netzwerktest

Mit dem Programm NTTCP kann man die Netzwerkgeschwindigkeit testen. Dazu wird auf einer Seite ein Server gestartet und auf einer anderen Seite ein entsprechende Client.

Den Server startet man durch Eingabe von nttcp -i -v. Der Server wartet dann auf eine Testanforderung des Clients. Um jetzt z.B die Geschwindigkeit zu testen gibt man auf dem Client nttcp -t <IP Adresse des Servers> ein.

So sieht ein gestarteter nttcp Server aus:

```
fli4l-server 3.6.2~# nttcp -i -v
nttcp-l: nttcp, version 1.47
nttcp-l: running in inetd mode on port 5037 - ignoring options beside -v and -p
```

So sieht ein Test mit einem nttcp Client aus:

```
fli4l-client 3.6.2~# nttcp -t 192.168.77.77
l~~8388608~~~~4.77~~~~0.06~~~~~14.0713~~~1118.4811~~~~2048~~~~429.42~~~34133.3
1~~8388608~~~~4.81~~~~0.28~~~~~13.9417~~~~239.6745~~~~6971~~~1448.21~~~24896.4
```

Die Hilfeseite von nttcp zeigt alle weiteren Parameter:

```
Usage: nttcp [local options] host [remote options]
       local/remote options are:
        -t      transmit data (default for local side)
        -r      receive data
        -l#     length of bufs written to network (default 4k)
        -m      use IP/multicasting for transmit (enforces -t -u)
        -n#     number of source bufs written to network (default 2048)
```

```
-u      use UDP instead of TCP
-g#us   gap in micro seconds between UDP packets (default 0s)
-d      set SO_DEBUG in sockopt
-D      don't buffer TCP writes (sets TCP_NODELAY socket option)
-w#     set the send buffer space to #kilobytes, which is
        dependent on the system - default is 16k
-T      print title line (default no)
-f      give own format of what and how to print
-c      compares each received buffer with expected value
-s      force stream pattern for UDP transmission
-S      give another initialisation for pattern generator
-p#     specify another service port
-i      behave as if started via inetd
-R#     calculate the getpid()/s rate from # getpid() calls
-v      more verbose output
-V      print version number and exit
-?      print this help
-N      remote number (internal use only)
default format is: %9b%8.2rt%8.2ct%12.4rbr%12.4cbr%8c%10.2rcr%10.1ccr
```

OPT_RTMON Installiert ein tool, dass Änderungen der Routingtabelle überwacht. Primäre Verwendung: Debugging

OPT_SERIAL Die Programme "stty" und "setserial" werden zum Konfigurieren der seriellen Schnittstellen installiert.
Diese Programme werden von anderen Paketen benutzt und sind des weiteren für Tests vorgesehen.

OPT_SHRED Installiert das Program *shred* auf dem Router, ein Programm zum gründlichen Löschen von Blockgeräten.

OPT_STRACE debug

Mit dem Programm strace können die Funktionsaufrufe, der Ablauf eines Programmes beobachtet werden

strace <programm>

OPT_TCPDUMP debug

Mit dem Programm tcpdump kann Netzwerkverkehr beobachtet, ausgewertet mitgeschnitten werden. Mehr dazu unter z.B. Google mit den Suchworten "tcpdump man"

tcpdump <parameter>

OPT_TRACEROUTE Verfolgen von Netzwerkrouten im Internet

Für Internet Protocol Version 4 (IPv4).

Bei dieser Version von traceroute ist es unbedingt notwendig das Interface mit an zu geben. Beispiel:

```
traceroute -i ppp0 www.fli4l.de
```

OPT_TRACEROUTE6 Verfolgen von Netzwerkrouten im Internet

Für Internet Protocol Version 6 (IPv6).

OPT_VALGRIND Installiert Valgrind auf dem Router.

OPT_WGET http/ftp Client

Mit dem Programm wget können Daten von einem Webserver im Batch abgerufen werden. Praktisch ist aber (und deswegen ist wget im fli4l-Paket dabei), daß man damit Umlenkungen des Providers auf den eigenen Webserver nach einem Verbindungsaufbau auf einfache Weise abfangen kann, z.B. für Freenet. Wie das geht, hat Steffen Peiser in einem Mini-HowTo erklärt.

Siehe: http://www.fli4l.de/hilfe/howtos/einsteiger/wget-und-freenet/

WGET_SSL Trägt man hier 'yes' ein, wird ein ssl-tüchtiges wget installiert. Zu beachten ist, dass das wget mit ssl *sehr* viel grösser ist als das ohne ssl.

Standardeinstellung: `WGET_SSL`='no'

OPT_YTREE Datei-Manager

Installiert Datei-Manager Ytree auf dem Router.

4.22. UMTS - Anbindung mittels UMTS an das Internet

Anbindung eines FLI4L mittels UMTS an das Internet. Für den Betrieb sind unter anderem auch weitere optionale Pakete erforderlich.

4.22.1. Konfiguration

OPT_UMTS Standard-Einstellung: `OPT_UMTS`='no'

'yes' aktiviert das Paket.

UMTS_DEBUG Standard-Einstellung: `UMTS_DEBUG`='no'

Soll pppd zusätzliche Debug-Informationen ausgeben, muss man `UMTS_DEBUG` auf 'yes' setzen. In diesem Fall schreibt pppd zusätzlichen Informationen über die syslog-Schnittstelle.

WICHTIG: Damit diese auch über syslogd ausgegeben werden, muss die Variable `OPT_SYSLOGD` (s.o.) ebenso auf 'yes' gesetzt sein.

UMTS_PIN Standard-Einstellung: `UMTS_PIN`='disabled'

Pin für die SIM-Karte

Erlaubt sind eine 4stellige Nummer oder das Wort 'disabled'

UMTS_DIALOUT Standard-Einstellung: `UMTS_DIALOUT`='*99***1#'

Wählparameter zum Herstellen der Verbindung

Einwahldaten einiger deutscher Netzbetreiber/Provider

Anbieter	APN	Benutzername	Passwort
T-Mobile	internet.t-mobile	beliebig	beliebig
Vodafone	web.vodafone.de	beliebig	beliebig
E-Plus	internet.eplus.de	eplus	gprs
O2 (Vertragskunden)	internet	beliebig	beliebig
O2 (Prepaid-Kunden)	pinternet.interkom.de	beliebig	beliebig
Alice	internet.partner1	beliebig	beliebig

UMTS_GPRS_UMTS Standard-Einstellung: `UMTS_GPRS_UMTS`='both'

Welche Übertragungsart soll genutzt werden

Erlaubte Werte (both, gprs, umts)

UMTS_APN Standard-Einstellung: `UMTS_APN`='web.vodafone.de'

UMTS_USER Standard-Einstellung: `UMTS_USER`='anonymer'

UMTS_PASSWD Standard-Einstellung: `UMTS_PASSWD`='surfer'

Hier werden die für die Einwahl nötigen Daten angegeben.

Es sind Benutzerkennung und Passwort für den jeweils benutzten Provider anzugeben. `UMTS_USER` enthält die Benutzerkennung, `UMTS_PASSWD` das Passwort.

Für einige deutsche Netzbetreiber/Provider lauten die APNs (Einwahlknoten)

- http://www.teltarif.de/mobilfunk/internet/einrichtung.html

UMTS_NAME Standard-Einstellung: `UMTS_NAME`='UMTS'

Hier sollte ein Name für den Circuit vergeben werden - max. 15 Stellen lang. Dieser wird im imon-Client imonc angezeigt. Leerstellen (Blanks) sind nicht erlaubt.

UMTS_HUP_TIMEOUT Standard-Einstellung: `UMTS_TIMEOUT`='600'

Hier kann die Zeit in Sekunden angegeben werden, nach welcher die Verbindung beendet werden soll, wenn nichts mehr über die UMTS-Verbindung läuft. Dabei steht ein Timeout von '0' für kein Timeout.

UMTS_TIMES Standard-Einstellung: `UMTS_TIMES`='Mo-Su:00-24:0.0:Y'

Die hier angegebenen Zeiten bestimmen, wann dieser Circuit aktiviert werden soll und wann er wieviel kostet. Dadurch wird es möglich, zu verschiedenen Zeiten verschiedene Circuits mit Default-Routen zu verwenden (Least-Cost-Routing). Dabei kontrolliert der Daemon imond die Routen-Zuweisung.

UMTS_CHARGEINT Standard-Einstellung: `UMTS_CHARGEINT`='60'

Charge-Interval: Hier ist der Zeittakt in Sekunden anzugeben. Dieser wird dann für die Kosten-Berechnung verwendet.

UMTS_USEPEERDNS Standard-Einstellung: `UMTS_USEPEERDNS`='yes'

Soll der DNS des Providers verwenden werden.

UMTS_FILTER Standard-Einstellung: `UMTS_FILTER='yes'`

Fli4l legt automatisch auf, wenn während der über hangup timeout angegebenen Zeit keine Daten über das ppp0-Interface gehen. Leider wertet das Interface auch Datentransfers mit, die von außen kommen, z.B. durch Verbindungsversuche eines P2P-Clients wie eDonkey. Da man heutzutage eigentlich pemanent von anderen kontaktiert wird, kann es passieren, daß fli4l die UMTS-Verbindung nie beendet.

Hier hilft die Option `UMTS_FILTER`. Setzt man es auf yes, wird nur noch Verkehr gewertet, der von der eigenen Maschine generiert wird und externer Traffic wird komplett ignoriert. Da von draußen reinkommender Traffic in der Regel dazu führt, daß der Router oder dahinter liegende Rechner reagieren, indem sie z.B. Verbindungswünsche ablehnen, werden zusätzlich noch einige rausgehende Pakete ignoriert.

UMTS_ADAPTER (optional)

Hier wird eingetragen ob es sich um eine PCMCIA-Karte, einen USB-Aadpter oder um ein per USB-Kabel angeschlossenes Telefon handelt.

Bei nichtvorhandensein der Variable werden nur die benötigten Dateien für einen USB-Adapter kopiert.

Erlaubte Werte: (pcmcia,usbstick,usbphone)

Alle folgenden Variablen sind optional und nur notwendig wenn die automatische Erkennung versagt.

UMTS_IDVENDOR (optional) `UMTS_IDVENDOR='xxxx'`

Hersteller ID nach Einschalten des Adapters

UMTS_IDDEVICE (optional) `UMTS_IDDEVICE='xxxx'`

Produkt ID nach Einschalten des Adapters

Angabe der folgenden beiden Parameter nur notwendig, wenn sich eine ID ändert nach der Initialisierung

UMTS_IDVENDOR2 (optional) `UMTS_IDVENDOR2='xxxx'`

Hersteller ID nach Initialisierung des Adapters

UMTS_IDDEVICE2 (optional) `UMTS_IDDEVICE2='xxxx'`

Produkt ID nach Initialisierung des Adapters

UMTS_DRV (optional) `UMTS_DRV='xxxx'`

Treiber zum Ansteuern Adapters, wenn nicht angegeben wird 'usbserial' genommen

UMTS_SWITCH (optional) `UMTS_SWITCH='-v 0x0af0 -p 0x6971 -M 555...000 -s 10'`

Parameter für usb-modeswitch zum Initialisieren des Modems. (siehe Website usb-modeswitch) Es sollten bis auf wenige Ausnahmen alle auf der Website genannten Modems automatisch erkannt werden.

- http://www.draisberghof.de/usb_modeswitch/

UMTS_DEV (optional)

Bei Problemen kann hier die Datenschnittstelle für den pppd angegeben werden. Für die Adapter sind das meist folgende:

```
ttyUSB0 für usbstick
ttyS2   für pcmcia
ttyACM0 für usbphone
```

UMTS_CTRL (optional)

Einige Adapter haben mehrere Schnittstellen, über denen das Modem gesteuert wird. Ist nur eine vorhanden können Statusinformationen nur im 'Offline'-Zustand ausgelesen werden. Bei einer Option Fusion UMTS Quad lautet die Schnittstelle zB: ttyUSB2.

4.22.2. Beispielkonfiguration für RRDTOOL

Je nach Hardware ist es möglich, über OPT_HTTPD (Seite 131) die Signalstärke und Bitfehler anzeigen zu lassen. Ausserdem kann der Verlauf der Signalstärke bzw. Bitfehlerrate mittels OPT_RRDTOOL aufgezeichnet werden. Bei mancher Hardware ist es nicht so sinnvoll, da Statusinformationen nur während des Offline-Zustandes ausgelesen werden können. Als Quelle für RRD ist dabei 'umts' anzugeben.

Beispielkonfig für rrdtool:

```
RRDTOOL_x_SOURCE='umts'
RRDTOOL_x_LABEL='UMTS Status'
RRDTOOL_x_OPTIONS_N='2'
RRDTOOL_x_OPTIONS_1='signal'
RRDTOOL_x_OPTIONS_1_LABEL='Signalstärke'
RRDTOOL_x_OPTIONS_2='error'
RRDTOOL_x_OPTIONS_2_LABEL='Bitfehler'
```

4.23. USB - Support für USB-Geräte

OPT_USB Hier wird die grundsätzliche Unterstützung von USB-Geräten ein- beziehungsweise ausgeschaltet. Erst wenn hier 'yes' eingetragen wird, können USB-Geräte überhaupt verwendet werden. Sollten Sie also in der base.txt ein USB-Gerät ausgewählt haben, so müssen Sie hier zwingend 'yes' eintragen. Andernfalls wird das Gerät nicht verwendet.

Standard-Einstellung: `OPT_USB='no'`

USB_LOWLEVEL Mit dieser Variable wird der Lowlevel-Treiber für die USB-Unterstützung ausgewählt. Der Häufigste ist der 'uhci'. Sollte dieser nicht funktionieren, oder der Router sich beim Laden des Moduls aufhängen, so probieren Sie die beiden anderen Module. Es gibt drei mögliche Einstellungen: 'uhci', 'ohci' und 'ehci'. Wenn man keine Platzprobleme hat (beispielsweise bei einer HD-Installation), ist es auch möglich mehrere Treiber anzugeben. Beim Booten sieht man dann, welcher Treiber tatsächlich geladen werden kann

und bei welchen es Fehlermeldungen gibt. Nach diesem 'Probelauf' sollte man aber hier den richtigen Treiber eintragen.

Bei Verwendung eines WRAP-Boards ist 'ohci' der Treiber der Wahl.

Standard-Einstellung: `USB_LOWLEVEL`='uhci'

USB_EXTRA_DRIVER_N Anzahl der zusätzlich zu ladenden Treiber.

Standard-Einstellung: `USB_EXTRA_DRIVER_N`='0'

USB_EXTRA_DRIVER_x Treiber der geladen werden soll.

Mögliche Werte im Momment

- keyboard - Unterstützung für USB-Tastaturen
- printer - Unterstützung für USB-Drucker
- belkin_sa - USB Belkin Serial converter
- cyberjack - REINER SCT cyberJack pinpad/e-com USB Chipcard Reader
- digi_acceleport - Digi AccelePort USB-2/USB-4 Serial Converter
- empeg - USB Empeg Mark I/II
- ftdi_sio - USB FTDI Serial Converter
- io_edgeport - Edgeport USB Serial
- io_ti - Edgeport USB Serial
- ipaq - USB PocketPC PDA
- ir-usb - USB IR Dongle
- keyspan - Keyspan USB to Serial Converter
- keyspan_pda - USB Keyspan PDA Converter
- kl5kusb105 - KLSI KL5KUSB105 chipset USB->Serial Converter
- kobil_sct - KOBIL USB Smart Card Terminal (experimental)
- mct_u232 - Magic Control Technology USB-RS232 converter
- omninet - USB ZyXEL omni.net LCD PLUS
- pl2303 - Prolific PL2303 USB to serial adaptor
- visor - USB HandSpring Visor / Palm OS
- whiteheat - USB ConnectTech WhiteHEAT

Standard-Einstellung: `USB_EXTRA_DRIVER_x`=''

USB_EXTRA_DRIVER_x_PARAM Parameter für den zusätzlichen Treiber. Im Normalfall muss hier nichts eingegeben werden.

Standard-Einstellung: `USB_EXTRA_DRIVER_x_PARAM`=''

USB_MODEM_WAITSECONDS Standard-Einstellung: `USB_MODEM_WAITSECONDS`='21'

Leider brauchen das Eagle und das Speedtouch USB Modem eine halbe Ewigkeit bis sie bereit sind. In den meisten Fällen reichen die 21 Sekunden, die als Standardeinstellung

genommen werden, für die Initialisierung aus. Manchmal hat man das Glück das man den Wert auch halbieren kann und sein Eagle oder Speedtouch USB Modem bereits nach 10 Sekunden einsatzbereit ist, dann kann man hier halt 10 Sekunden eintragen. Wenn man Pech hat muss man den Wert erhöhen. Hier hilft leider nur probieren und austesten.

4.23.1. Probleme mit USB-Geräten

Es kann bei einigen USB-Geräten zu Problemen kommen. Das kann verschiedene Ursachen haben, wie Beispielsweise der Treiber-Software oder dem USB-Controller.

In der vorliegenden Version funktioniert das Eagle-USB-DSL-Modem nur dann, wenn es auch an den Splitter angeschlossen ist. Sollte das nicht der Fall sein, wird das entsprechende eth-Device nicht generiert. Das hat zur Folge, das das Modem nicht verwendbar ist. Schliessen Sie also bitte vorher das Modem an die Telefonleitung an.

4.23.2. Hinweise zur Benutzung

Es ist darauf zu achten, dass die USB-Unterstützung hardwareseitig aktiviert ist. Insbesondere bei Onboard-USB-Kontrollern ist das wichtig. So wird z. B. ein WRAP ohne USB-Anschluss ausgeliefert. USB kann hier durch ein Zusatzmodul nachgerüstet werden und ist aus diesem Grund im BIOS standardmäßig deaktiviert.

4.23.3. Mounten von USB-Geräten

Eingesteckte USB-Geräte werden zwar automatisch erkannt, müssen aber 'von Hand' sowohl an- als auch abgemeldet werden. Beim Einstecken z. B. eines USB-Stick wird dieser als SCSI-Device erkannt. Aus diesem Grund erfolgt der Zugriff über das Device sd# bei SuperFloppy-Geräten bzw. über sd#<Partitionsnummer> bei Geräten mit einer Partitionstabelle. USB-Sticks werden wie Festplatten behandelt, also bei zwei USB-Anschlüssen sda1 und sdb1 angesprochen. USB-Floppies hingegen werden durch sda bzw. sdb angesprochen, also ohne Angabe einer Partitionsnummer.

Somit kann ein USB-Stick durch das Kommando
mount /dev/sda1 /mnt
nach /mnt gemountet werden. Analog dazu durch
mount /dev/sdb1 /mnt
für das zweite USB-Gerät. Die Geräte werden in der Reihenfolge des Einsteckens benannt, also erstes USB-Gerät = sda, zweites USB-Gerät = sdb etc. pp. Es lässt sich somit nicht fix definieren, welcher der USB-Ports welche 'Bezeichnung' hat, da diese von der Reihenfolge des Einsteckens der Geräte abhängt. Die Abmeldung der angemeldeten USB-Geräte erfolgt durch
umount /mnt

Bei gleichzeitiger Verwendung mehrerer USB-Geräte sollte es unbedingt vermieden werden, alles in ein Ziel zu mounten. Aus diesem Grund bietet es sich an, unterhalb von /mnt weitere Verzeichnisse anzulegen, in welche die Geräte dann gemountet werden können. Dies kann z. B. wie folgt erledigt werden:

mkdir /mnt/usba mkdir /mnt/usbb

Beim Mounten der Geräte werden dann diese Verzeichnisse als Ziel angegeben:

mount /dev/sda1 /mnt/usba mount /dev/sdb1 /mnt/usbb

Somit ist der Inhalt der USB-Geräte unter /mnt/usba bzw. /mnt/usbb zu finden. Die Abmeldung erfolgt dann durch

umount /mnt/usba umount /mnt/usbb

Wenn mehrere Partitionen je USB-Gerät existieren, müssen die Verzeichnisse unterhalb von /mnt entsprechend strukturiert werden.

4.24. WLAN - Wireless-LAN Unterstützung

4.24.1. Hinweise zum Prism54 Treiber

Der Treiber für Prism54 basierte WLAN-Karten ist noch recht jung und daher noch nicht 100% stabil. Gerade bei schlechtem Empfang kommt es ab und zu vor, dass Karten mit dem Prism54 Treiber einen ethX Transmit Error produzieren. In diesem Fall hilft nur ein Neustart des fli4l Routers. Im der fli4l Version 2.1.12 wird der Prism54 Treiber vom 27.02.2004 benutzt.

Achten Sie in jedem Fall darauf, dass Sie beim Einsatz von PCI Karten ein Mainboard benutzen, was mindestens die PCI 2.2 Spezifikationen erfüllt. Auf älteren Mainboard die nur PCI 2.1 oder älter unterstützen kann es zu den unterschiedlichsten Fehler kommen. Entweder startet der Computer gar nicht (er läßt sich nicht einmal einschalten), oder die WLAN-Karte wird beim PCI Scan nicht gefunden.

Ein weiteres Problem ist, dass es von einigen Prism54 basierten WLAN-Karten verschiedene Hardwareversionen gibt. Die neueren (v2 Version genannt) Karten werden aktuell *nicht* vom Prism54 Treiber unterstützt.

WLAN-Karten werden in der base.txt `IP_NET_X_DEV` mit wlanX angesprochen. Wenn nur eine WLAN-Karte im System ist, hat diese also den Namen wlan0.

4.24.2. WLAN-Konfiguration

OPT_WLAN Standard-Einstellung: `OPT_WLAN`='no'

Aktiviert das Wireless LAN Option Pack.

WLAN_WEBGUI Standard-Einstellung: `WLAN_WEBGUI`='yes'

Aktiviert das Webinterface für das Wireless LAN Option Pack.

WLAN_REGDOMAIN (optionale Variable)

Mit dieser Variable kann man die Landesspezifischen Einstellungen anpassen. Gültige Werte sind ISO 3166-1 alpha-2 Ländercodes wie z.B. 'DE' In verschiedenen Ländern gelten verschiedene Vorgaben für die Kanalauswahl und Sendeleistungen.

WLAN_N Anzahl der voneinander unabhängigen WLAN-Konfigurationen. Steht hier eine '1' so ist das Verhalten wie in früheren Versionen von Fli4l.

WLAN_x_MAC MAC-Adresse der WLAN-Karte in dieser Schreibweise:

XX:XX:XX:XX:XX:XX

Jedes X ist ein Hex-Digit der Mac-Adresse der Karte, für die diese Konfiguration gelten soll. Sollte keine der hier eingetragenen Mac-Adressen zu einer Karte passen, so wird die Konfiguration `WLAN_1_*` auf diese Karte angewandt und es wird eine Warnmeldung ausgegeben, die auf den Umstand hinweist. Die Warnmeldung enthält die festgestellte MAC-Adresse der Karte. Diese Meldung erscheint jedoch nur dann, wenn `WLAN_N` größer '1' ist. Ist `WLAN_N` gleich '1' so ist der Inhalt dieser Variable bedeutungslos.

WLAN_x_MACOVERRIDE Ändert die MAC-Adresse der WLAN-Karte damit man als Client an ein WLAN mit MAC-Filter verbinden kann ohne dort den Filter anpassen zu müssen. Hilfreich bei WAN-Anbindungen, die z.B. auf die MAC-Adresse eines gelieferten WLAN-USB-Sticks gebunden sind.

WLAN_x_ESSID Die SSID ist der Name für das Funknetzwerk. Die auch "Network Name" genannte Zeichenfoge kann bis zu 32 Zeichen lang sein. Sie wird im AP eines WLAN konfiguriert und von allen Clients, die darauf Zugriff haben sollen, eingestellt. Auch bei Ad-Hoc muß die SSID auf allen teilnehmenden Nodes identisch sein.

WLAN_x_MODE Stellt den zu verwendenden WLAN-Modus der Karte ein.

Standard-Einstellung: `WLAN_x_MODE`='ad-hoc'

Mögliche Werte:

ad-hoc	für ein Funknetz ohne Access-Point
managed	gemanagedtes Funknetz mit mehreren Zellen
master	die WLAN-Karte arbeitet als Access-Point

`WLAN_x_MODE`='master' funktioniert nur mit einem geeigneten WLAN-Treiber. Zur Zeit ist das lediglich mit den Treibern hostap_(cs|pci|plx) ,prism54 und ath_pci möglich. Externe Treiber können diesen Modus auch unterstützen, Bitte lesen sie hierzu die Dokumentation des Treibers.

WLAN_x_NOESSID Ermöglicht das Abschalten der ESSID in den Beacon Frames. Nur möglich mit Treiber hostap_* und Firmware >= 1.6.3 im WLAN_MODE='master'

Dieses Feature ist optional und muß manuell zur config/wlan.txt hinzugefügt werden.

WLAN_x_CHANNEL Setzt den Übertragungskanal des Netzwerks.

Standard-Einstellung: `WLAN_x_CHANNEL`='1'

Mögliche Werte: 1-13 und 36,40,44,48,52,56,60,64,100,104,108,112,116,120,124,128,132,136,140

Bitte lesen sie die Dokumentation Ihrer WLAN-Karte um herauszufinden, welche Kanäle in Ihrem Land erlaubt sind. Sollten sie hier einen nicht erlaubten Kanal einstellen, so sind sie alleine dafür verantwortlich. In Deutschland sind die Kanäle 1-13 im Frequenzband 2,4 GHz (Modi: b und g) erlaubt. Die Kanäle im Bereich 36-140 (siehe oben) sind im 5 GHz zulässig.

Desweiteren ist der Wert '0' erlaubt, falls `WLAN_x_MODE`='managed' gesetzt ist. Dadurch wird kein expliziter Kanal eingestellt, sondern der AP auf allen verfügbaren Kanälen gesucht. Man kann dem Kanal-Wert auch einen Buchstaben a,b oder g anhängen (z.B. 5g), welcher dann den gewünschten Betriebsmodus/Frequenzband auswählt.

Ein angehängtes 'n' oder 'N' selektiert bei entsprechenden WLAN-Karten die Nutzung von 802.11n. Kleingeschrieben bedeutet: 20 MHz Kanalbreite, grossgeschrieben: 40 MHz Kanalbreite.

Großschreibung bei a/b/g sorgt bei einigen (aktuell nur ath_pci) Treibern dafür, daß proprietäre WLAN-Turbos aktiviert werden. Diese Option ist experimentell und kann auch wieder entfernt werden.

WLAN_x_RATE Setzt die Übertragungsgeschwindigkeit des Netzwerks.

Standard-Einstellung: `WLAN_x_RATE`='auto'

Mögliche Werte: 1,2,5.5,11,auto - Angaben in Megabit/s je nach Karte können auch noch diese Raten ausgewählt werden: 6,9,12,18,24,36,48 und 54. Bei manchen 54 MBit-Karten kann die Rate nicht angegeben werden. Hier ist dann 'auto' einzutragen.

WLAN_x_RTS Aktiviert RTS/CTS Handshake. Diese Option ist in grossen Wlans mit vielen sendenden Clients nuetzlich wenn sich die Clients gegenseitig nicht hoeren koennen sondern nur den AP. Ist diese Option aktiviert sendet der Client vor jedem Sendevorgang ein RTS mit der Bitte um Erlaubnis zum Senden und bekommt ein CTS, die Erlaubnis zum Senden, vom AP zurueck. Damit weiss jeder Client dass ein Client sendet auch wenn er diesen Client nicht hoert. Hierdurch werden Kollisionen vermindert weil sicher gestellt ist dass immer nur ein Client sendet. Diese Option macht nur unter der oben beschriebenen Situation Sinn weil sie zusaetzlichen overhead hinzufuegt und somit die Gesamtbandbreite verringert. Durch die Verringerung von Kollisionen kann sich die Bandbreite jedoch wieder erhoehen.

Dieses Feature ist optional und muß manuell zur config/wlan.txt hinzugefügt werden.

WLAN_x_ENC_N Legt die Anzahl der Wireless Encryption Key's fest.

Mögliche Werte: 0-4

WLAN_x_ENC_x Setzt die Wireless Encryption Keys.

Mögliche Werte:

XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XX	128 Bit Hex-Key (X=0-F)
XXXX-XXXX-XX	64 Bit Hex-Key (X=0-F)
s:<5 Zeichen>	64 Bit
s:<6-13 Zeichen>	128 Bit
P:<1-64 Zeichen>	128 Bit

Das Verfahren der Key-Vergabe mit s:Text ist _nicht_ mit der Passphrase der Windows-Treiber kompatibel. Hier bitte einen Hex-Key verwenden! Unter Windows wird der Hex-Key meist _ohne_ die Bindestriche '-' verwendet. Die Angabe mittels P:<Text> ist kompatibel zur Passphrase der meisten Windows WLAN-Treiber (wenn nicht allen) aber _nur_ im 128 Bit Modus. Linux erlaubt es, verschiedene Schlüssellängen zu mischen. Windows-Treiber jedoch in der Regel _nicht_!

WLAN_x_ENC_ACTIVE Legt den aktiven Wireless Encryption Key fest.

Mögliche Werte: 1-4

Diese Variable ist aufzunehmen, wenn `WLAN_x_ENC_N` > 0 gesetzt wird. Ansonsten optional.

WLAN_x_ENC_MODE Aktiviert den Encryption Mode.

Mögliche Werte:

on/off	mit oder ohne Veschlüsselung
open	nimmt auch unverschlüsselte Pakete an
restricted	nimmt nur veschlüsselte Pakete an

Sinnvoller Wert: 'restricted'

Dieses Feature ist optional und muß manuell zur config/wlan.txt hinzugefügt werden. Ist die Variable nicht vorhanden, so wird als Default 'off' angenommen, wenn kein WEP-Key definiert ist und 'restricted' wenn mindestens ein Key definiert ist.

WLAN_x_WEP_ROTATE Legt fest nach wie vielen Sekunden der WEP Key gewechsellt werden soll.

Mögliche Werte: 0 - 1800 oder 1r - 1800r

0 deaktiviert die Funktion (default) 1-1800 die Anzahl Sekunden zwischen den Wechseln Das optionale 'r' führt zu einer Modulation des Wechselintervalls mit einem Zufallswert welches die vorhersagbarkeit der Wechselzeitpunktes erschweren sollte. Die Zahl gibt dann den durchschnittlichen Abstand zwischen 2 Wechseln an. Der tatsächliche Abstand liegt im Bereich von +- 1/2 des Wertes. Also bei 300r variiert der Abstand zwischen 150 und 450 Sekunden. Für die Funktion dieser Option ist es nötig, mindestens 2 WEP-Keys zu definieren. Bei den Clients müssen alle hier definierten WEP-Keys in der selben Reihenfolge auch eingetragen werden. Windows XP kann hiermit nur bedingt umgehen, da die in Windows XP eingebaute WLAN-Funktionalität per Default nur einen Key erlaubt. Dieses Feature ist optional und muß manuell zur config/wlan.txt hinzugefügt werden.

WLAN_x_WPA_KEY_MGMT (Experimentell) Will man statt WEP-Verschlüsselung WPA verwenden, stellt man hier den gewünschten WPA-Modus ein. Momentan wird nur WPA-PSK unterstützt, also WPA mit einem Client und Access-Point vorab bekannten Schlüssel. Dieser Schlüssel sollte sorgfältig gewählt werden und nicht zu kurz sein, da er ansonsten auch anfällig gegen Wörterbuchattacken ist.

Unterstützt werden im *managed*-Mode alle vom Wpa-Supplicanten (http://hostap.epitest.fi/wpa_supplicant/ und im *master*-Mode alle vom Hostapd (http://hostap.epitest.fi/hostapd/) unterstützten Karten.

Erfolgreich getestet wurden bereits Karten basierend auf den Chipsätzen von Atheros und vom hostap-Treiber unterstützte Karten (sowohl im managed als auch im master mode). Theoretisch ist auch noch Unterstützung für atmel-Karten, Karten, die vom ndiswrapper angesteuert werden, und einige andere möglich. Hier müssen die Ersteller der entsprechenden Opts aber ihre Opt-Pakete noch entsprechend anpassen.

WLAN_x_WPA_PSK Hier wird der Schlüssel angegeben, der zur Kommunikation zwischen Client und Access-Point verwendet werden soll. Dieser Schlüssel wird in Form einer Passphrase (eines Satzes) angegeben, die mindestens 16 Zeichen lang sein muß und bis zu 63 Zeichen lang sein kann.

WLAN_x_WPA_TYPE (Experimentell) Zur Auswahl stehen hier 1 für WPA1, 2 für den WPA2 (IEEE 802.11i) Modus und 3 für beide - der Client kann dann entscheiden ob er WPA1 oder WPA2 nutzen möchte. Wenn die WLAN Hardware den Standard unterstützt, so ist dem sicheren WPA2 Verfahren den Vorzug zu geben.

WLAN_x_WPA_ENCRYPTION (Experimentell) Die Verschlüsselungsprotokolle TKIP und die erweiterte Version CCMP (AES-CTR/CBC-MAC Protocol, manchmal auch nur AES genannt) stehen hier zur Verfügung. CCMP wird eventuell nicht von älterer WLAN-Hardware unterstützt. Es können auch beide gemeinsam angegeben werden.

WLAN_x_WPA_DEBUG (Experimentell) Bei Problemen mit der WPA-Anbindung kann man diese Variable auf 'yes' setzen, um den zuständigen daemon zu umfangreicheren Ausgaben zu veranlassen. Diese kann man dann zur Diagnose der Probleme verwenden.

WLAN_x_AP Registriert diese Node bei einem Access-Point.

Hier ist die MAC-Adresse des Access-Points anzugeben. Wenn man bereits den WLAN-Mode "master" ausgewählt hat, ist dieser Eintrag leer zu lassen. Diese Option ist nur dann sinnvoll, wenn der fli4l den AP nicht von selber finden kann oder an einen bevorzugten Access-Point gebunden werden soll. Nur zum Einsatz in WLAN-Mode "managed" gedacht.

Dieses Feature ist optional und muß manuell zur config/wlan.txt hinzugefügt werden.

WLAN_x_ACL_POLICY Policy der ACL.

Standard-Einstellung: `WLAN_x_ACL_POLICY='allow'`

Beschreibt eine Aktion, der die angegebenen MAC-Adressen unterliegen:

deny	Keine der aufgelisteten MAC-Adressen erhält Zugang
allow	Alle aufgelisteten MAC-Adressen erhalten Zugang
open	Alle MAC-Adressen erhalten unabhängig vom Filter Zugang

Leider werden WLAN_ACL's aktuell nur von einem Treiber sauber unterstützt: hostap_* Als Alternative bieten sich die in 3.0.x deutlich erweiterten Firewall-Möglichkeiten an.

WLAN_x_ACL_MAC_N AP-ACLs - Einschränkung der erlaubten WLAN-Stationen.

Standard-Einstellung: `WLAN_x_ACL_MAC_N='0'`

Eine Zahl größer 0 aktiviert die Access Control List (der MAC-Adressenfilter) und gibt die Anzahl der ACL-Einträge an. Die Access Control List ist eine Liste von MAC-Adressen, denen der Zugang zum Access Point (AP) erlaubt/verboten wird. Anzahl der Mac-Adressen, die definiert werden.

WLAN_x_ACL_MAC_x Mac-Adressen in der Form: 00:00:E8:83:72:92

WLAN_x_DIVERSITY Hiermit kann man einstellen, ob die Antennen-Diversity aktiviert wird.

Standard-Einstellung: `WLAN_x_DIVERSITY='no'`

WLAN_x_DIVERSITY_RX Auswahl der Empfangsantenne.

Standard-Einstellung: `WLAN_x_DIVERSITY_RX='1'`

0 = Automatische Auswahl
1 = Antenne 1
2 = Antenne 2

WLAN_x_DIVERSITY_TX Auswahl der Sendeantenne.

Standard-Einstellung: `WLAN_x_DIVERSITY_TX='1'`

4.24.3. Beispiele

Anbindung an einen Access Point via WPA

```
OPT_WLAN='yes'
WLAN_N='1'
WLAN_1_MAC='00:0F:A3:xx:xx:xx'
WLAN_1_ESSID='foo'
WLAN_1_MODE='managed'           # Anbindung an Access Point
WLAN_1_CHANNEL='1'
WLAN_1_RATE='auto'
#
# WPA Konfiguration
#
WLAN_1_ENC_N='0'                # kein WEP
WLAN_1_WPA_KEY_MGMT='WPA-PSK'   # WPA pre shared key
WLAN_1_WPA_TYPE='1'             # WPA 1
WLAN_1_WPA_ENCRYPTION='TKIP'
WLAN_1_WPA_PSK='Deine gute Passphrase (16-63 Zeichen)'
#
# irrelevant im WPA Kontext
#
WLAN_1_ENC_N='0'
WLAN_1_ENC_ACTIVE='1'
WLAN_1_WEP_ROTATE='0'
WLAN_1_ACL_POLICY='allow'
WLAN_1_ACL_MAC_N='0'
```

Access Point mit WPA2 Verschlüsselung

```
OPT_WLAN='yes'
WLAN_N='1'
WLAN_1_MAC='00:0F:A3:xx:xx:xx'
WLAN_1_ESSID='foo'
WLAN_1_MODE='master'            # Access Point
WLAN_1_CHANNEL='1g'             # Channel 1, Modus 'g' auf einer
                                # Atheros-Karte
WLAN_1_RATE='auto'
#
# WPA Konfiguration
#
WLAN_1_ENC_N='0'                # kein WEP
WLAN_1_WPA_KEY_MGMT='WPA-PSK'   # WPA pre shared key
WLAN_1_WPA_TYPE='2'             # WPA 2
WLAN_1_WPA_ENCRYPTION='CCMP'
WLAN_1_WPA_PSK='Deine gute Passphrase (16-63 Zeichen)'
#
# MAC basierte Zugriffkontrolle auf AP
#
WLAN_1_ACL_POLICY='allow'
WLAN_1_ACL_MAC_N='0'
#
# irrelevant im WPA Kontext
```

```
#
WLAN_1_ENC_ACTIVE='1'
WLAN_1_WEP_ROTATE='0'
```

Access Point mit WEP Verschlüsselung und Key-Rotation

```
OPT_WLAN='yes'
WLAN_N='1'
WLAN_1_MAC='00:0F:A3:xx:xx:xx'
WLAN_1_ESSID='foo'
WLAN_1_MODE='master'            # Access Point
WLAN_1_CHANNEL='1'
WLAN_1_RATE='auto'
#
# WEP Konfiguration
#
WLAN_1_WPA_KEY_MGMT=''          # kein WPA
WLAN_1_ENC_N='4'                # 4 WEP-Keys
WLAN_1_ENC_1='...'
WLAN_1_ENC_2='...'
WLAN_1_ENC_3='...'
WLAN_1_ENC_4='...'
WLAN_1_ENC_ACTIVE='1'           # erster Schlüssel ist aktiv
WLAN_1_WEP_ROTATE='60'          # alle 60 Sekunden wird ein neuer
                                # ausgewählt
#
# MAC basierte Zugriffkontrolle auf AP
#
WLAN_1_ACL_POLICY='allow'
WLAN_1_ACL_MAC_N='0'
#
# irrelevant für WEP Konfiguration
#
WLAN_1_WPA_TYPE='2'
WLAN_1_WPA_ENCRYPTION='CCMP'
WLAN_1_WPA_PSK='...'
```

Client Anbindung an einen Access Point mit WEP Verschlüsselung und Key-Rotation

```
OPT_WLAN='yes'
WLAN_N='1'
WLAN_1_MAC='00:0F:A3:xx:xx:xx'
WLAN_1_ESSID='foo'
WLAN_1_MODE='managed'           # Access Point
WLAN_1_CHANNEL='1'
WLAN_1_RATE='auto'
#
# WEP Konfiguration
#
WLAN_1_WPA_KEY_MGMT=''          # kein WPA
WLAN_1_ENC_N='4'                # 4 WEP-Keys
WLAN_1_ENC_1='...'
```

```
WLAN_1_ENC_2='...'
WLAN_1_ENC_3='...'
WLAN_1_ENC_4='...'
WLAN_1_ENC_ACTIVE='1'           # erster Schlüssel ist aktiv
WLAN_1_WEP_ROTATE='60'          # alle 60 Sekunden wird ein neuer
                                # ausgewählt
#
# irrelevant für WEP Konfiguration
#
WLAN_1_ACL_POLICY='allow'
WLAN_1_ACL_MAC_N='0'
WLAN_1_WPA_TYPE='2'
WLAN_1_WPA_ENCRYPTION='CCMP'
WLAN_1_WPA_PSK='...'
```

4.24.4. rrdtool Integration

Mittels *rrdtool*, einem externen Paket, kann die Verbindungsqualität einer WLAN-Verbindung überwacht werden. Dies funktioniert aber nur im `WLAN_x_MODE='managed'` vollständig.

```
RRDTOOL_1_SOURCE='wlan'                 # Source
RRDTOOL_1_LABEL='WLAN'                  # Label (optional)
RRDTOOL_1_OPTIONS_N='1'                 # Number of WLANs
RRDTOOL_1_OPTIONS_1='wlan0'             # Device Number 1 = wlan0
RRDTOOL_1_OPTIONS_1_LABEL='Mein WLAN'   # Label (optional)
```

4.24.5. Spendenhinweis

Durch die großzügige Spende von 2 Ralink 2500 basierten WLAN-Karten können WLAN-Karten mit dem RT25xx Chipsatz mit Fli4l in den Modi ad-hoc und managed verwendet werden. Als Treiber ist in der base.txt hierzu rt2500 auszuwählen. Die Karten wurden gespendet von:

Computer Contor, Pilgrimstein 24a, 35037 Marburg

4.25. SRC - Das fli4l Buildroot

Dieses Kapitel ist nur für Entwickler interessant, die auch Binärprogramme oder Kernelmodule für fli4l compilieren wollen. Wenn Sie fli4l nur als Router einsetzen wollen und keine opt–Pakete für fli4l anbieten wollen, die eigene Kernelmodule oder Binärprogramme benötigen, können Sie dieses Kapitel komplett überspringen.

C++ ist im uClibc buildroot nicht enthalten. Es gibt zwar eine uClibc++, aber die befindet sich noch im Entwicklungsstadium. Und damit C++ überhaupt mit der uClibc funktioniert müssen bestimmte Optionen in der uClibc aktiviert werden. Unter anderem UCLIBC_CTOR_DTOR und UCLIBC_DYNAMIC_ATEXIT. Beides ist aus Platzgründen nicht aktiviert. Zudem muss noch ein C++ Compiler im Buildroot cross compiliert werden. Alles in allem viel Aufwand für einige wenige C++ Programme. C++ ist halt für Plattformen wie fli4l immer noch ein fettes Stück Software. Es ist auch nicht einfach möglich die beiden oben erwähnten Optionen einzuschalten und dann das Buildroot neu zu übersetzen. Einige

Programme die fli4l benötigt arbeiten noch nicht mit der uClibc++ zusammen. Wenn also wirklich jemand ein C++ Programm für den fli4l benötigt ist im Moment der einzige Weg das entsprechende Programm statisch zu übersetzen.

4.25.1. Eine Übersicht über die Sourcen

Im *src* Verzeichnis finden Sie folgende Unterverzeichnisse:

Verzeichnis	Inhalt
fbr2	In diesem Verzeichnis befindet sich ein angepaßtes Buildsystem, das auf der Toolchain zur uClibc 0.9.30.3 basiert. Damit ist es möglich das komplette fli4l Buildroot (=fbr) neu zu compilieren. Das ist in den meisten Fällen aber nicht notwendig, um neue opt- Pakete für fli4l zu entwickeln. Sie können ein Image des aktuellen fbr downloaden. Das Script setup-devsys.sh lädt dieses Image und setzt es automatisch für Sie auf.
fli4l	Die fli4l spezifischen Sourcen nach Paketen geordnet. Alle Sourcen, die in diesem Unterverzeichnis enthalten sind, wurden entweder speziell für die Verwendung mit fli4l geschrieben oder zumindest stark angepaßt.
buildroot	Das eigentliche fli4l Buildroot. Nach dem Vorbild der uClibc ist hier ein Buildsystem aufgebaut, mit dessen Hilfe die von fli4l verwendeten Programme neu gebaut werden können.
kernel-2.x	Die Scripte und Makefile zum Neucompilieren des fli4l Kernels. Mit allen Patches und externen Modulen.

4.25.2. Das fbr Image vorbereiten

Wer als Entwickler schnell einsteigen will, sollte sich das fertig compilierte und direkt benutzbare fli4l-Buildroot (in Zukunft fbr genannt) installieren. Diese Entwicklungsumgebung entspricht zu fast 100% der vom fl4il–Team intern eingesetzten. Lediglich einige Dateien wie etc/passwd, etc/group usw. entsprechen nicht dem, was auf dem Entwicklungsserver benutzt wird.

Kurzversion

Führen Sie das Script setup-devsys.sh aus, z.B. wie folgt:

```
cd src/fbr2
sh setup-devsys.sh
```

Das Script benötigt teilweise root-Rechte, die es nicht hat und gibt daher einige Kommandos aus, die man nach Ende des Scriptes noch ausführen muß, um die Installation zu vervollständigen.

Langversion

- Bitte downloaden Sie das fbr Image 'dev-sys-<datum>.tar.bz2' von http://source-archives.nettworks.org/ und speichern den Download auf Ihrem lokalen Computer ab. Die aktuelle Version finden Sie in der Datei src/fbr2/fbr-versions (momentan 29-03-2010)
- Entpacken Sie das Archiv. Beim Entpacken werden einige Dateien nicht angelegt werden können (*/dev/{console,null,randon,urandom,zero}*; diese müssen Sie nachträglich wie folgt anlegen:

```
cd fli4l_dev_sys/dev
mknod null    1 3
mknod random  1 8
mknod urandom 1 9
```

- Sie sollten jetzt noch die Dateien /etc/passwd und /etc/group in das fbr kopieren. Wenn Sie in /etc/group eine grosse Anzahl von Benutzer bei den einzelnen Gruppen haben, kann es sein, dass die Gruppen nicht korrekt im fbr angezeigt werden. Dies ist ein kosmetischer Bug der BusyBox.

Betreten der Buildroot-Umgebung — chroot

Zum Betreten der Build-Umgebung muss man in eine andere Umgebung mit eigener Libc wechseln. Dies geschieht in der Regel über den Befehl chroot. Diesen kann man nur als Root (privilegierter Nutzer) ausführen, was zu einigen Problemen führt. Man kann chroot als Root ausführen, dann arbeitet man allerdings innerhalb der Build-Umgebung als Root. Das kann man etwas entschärfen, indem man chroot die in neueren Versionen erlaubten Parameter –user-spec mitgibt und dem Nutzer per *sudo* den Aufruf von chroot erlaubt. In einer Umgebung, in der man selber Root ist, ist das unproblematisch. In einer Umgebung, in der man den Nutzern nicht traut, ist das nicht so einfach. Daher bringt fli4l ein eigenes kleines Tool mit, dass diese Probleme löst. Es wird als setuid-Programm installiert (erhält beim Starten root-Rechte, verwirft diese abersofort nach Ausführen des chroot-Systemrufes und wechselt damit auf die Id des aufrufenden Nutzers. Übersetzt und installiert wird das Tool wie folgt:

```
gcc -o $HOME/bin/uchroot $base/source/uchroot.c
sudo chown root $HOME/bin/uchroot
sudo chmod u+s $HOME/bin/uchroot
```

Das Script *src/buildroot/toolchain/enter_fbr.sh* kann dann genutzt werden, um die Buildumgebung zu betreten. Es nimmt dabei automatisch die zur Verfügung stehende Variante, also entweder *uchroot* oder *sudo chroot –user-spec ....*

4.25.3. Mit dem fbr arbeiten

Um jetzt mit dem fbr zu arbeiten müssen Sie das Script *enter_fbr.sh* ausführen. Sie befinden sich dann innerhalb der Entwicklungsumgebung, die Sie benötigen, um Programme für den fli4l Router zu compilieren. Innerhalb der fbr Umgebung können Sie die Programme wie gewohnt mit ./configure und make einrichten und übersetzen. Die meisten Programme laufen ohne jede

Änderung direkt in der fbr Umgebung für die ersten Tests. Die Entwicklungsumgebung ist sehr rudimentär, man sollte sie also wirklich nur zum Übersetzen verwenden. Typischerweise arbeitet man dann mit zwei Umgebungen, in der normalen editiert man und nutzt die gewohnten tools, in der anderen compiliert man das ganze dann. Wenn man das ganze dann irgendwann automatisieren will, um möglichst leicht auf neue Versionen wechseln zu können, kann man das Paket analog zu den fli4l Binaries bauen.

4.25.4. Die Programme der fli4l-Distribution übersetzen

Im Verzeichnis src/buildroot finden Sie ein Makefile, welches die Übersetzung aller Programme aus den Basispaketen für den fli4l Router steuert. Wenn Sie innerhalb der fbr Umgebung in diesem Verzeichnis ein make ausführen, dauert es je nach verwendetem Rechner und Internetanbindung zwischen einer Stunde bis hin zu mehreren Stunden und Sie finden dann in *src/buildroot/target/<benutzername>* alle übersetzten Programme. Die Übersetzung der einzelnen Pakete wird über kleine Makefiles gesteuert. Wie diese aufgebaut sind und wie man daraus abgeleitet eigene Makefiles schreiben kann, wird hier beschrieben: http://buildroot.uclibc.org/buildroot.html#add_software.

Dann kann man in einem ersten Schritt die benötigten Quellen beschaffen, um sicherzustellen, daß das Bauen nicht wegen falscher Links abgebrochen wird. Die Links werden teilweise ungültig, wenn neuere Versionen der verwendeten Programme erscheinen; sie müssen dann angepasst werden. Um die Quellen zu beschaffen, führt man folgendes aus:

```
cd <pfad des buildroots>/home/<user name>/fli4l/src/buildroot
make source
```

Nun nur noch fbr eingeben, um im Buildroot zu landen und dann make um die Programme zu generieren.

4.25.5. Eigene Programme ins fbr einbinden

Die Programme werden mit Hilfe von Makefiles übersetzt, die nach dem Vorbild des *mu*Clibc-Buildroot (http://buildroot.uclibc.org/buildroot.html#add_software) geschrieben wurden. Will man seinem Opt entsprechende Makefiles beilegen, sollte man diese unter src/buildroot/package/<package name>/ ablegen. Beispiele findet man unter src/buildroot/package/templates. Zu beachten ist dabei, daß bei fli4l noch keine BR2_PACKAGE_FOO-Variablen existieren, man die TARGETS+= Anweisung also ohne if drumherum einfügen muß.

Eigene Programme in der fli4l-Repository-Struktur

Die Repository-Struktur weicht leicht von der Struktur der veröffentlichten Pakete ab. Will in diesen Paketen auf Patches oder ähnliches Zugreifen, muß man dies über einen Prefix tun, den man wie folgt bestimmen kann (foo steht für den Paket-Namen):

```
ifneq ($(FLI4L_DIR),)
FOO_PREFIX:=$(FLI4L_DIR)/foo/src/buildroot/
endif
```

4.25.6. Der Kernel

Der Kernel wird ganz normal mit dem lokal installierten Compiler gebaut, man benötigt dazu nicht den Compiler des Buildroots. Der fli4l Kernel wird vom Entwicklerteam mit dem gcc-3.4 von Debian Etch übersetzt. Eine Beschreibung des Kernelbaus ist in der Entwickler-Dokumentation zu finden.

```
Reading specs from /usr/lib/gcc/i486-linux/3.4.4/specs
Configured with: ../src/configure -v
--enable-languages=c,c++,java,f77,pascal,objc,ada,treelang
--prefix=/usr --libexecdir=/usr/lib
--with-gxx-include-dir=/usr/include/c++/3.4 --enable-shared
--with-system-zlib --enable-nls --without-included-gettext
--program-suffix=-3.4 --enable-__cxa_atexit
--enable-libstdcxx-allocator=mt --enable-clocale=gnu
--enable-libstdcxx-debug --enable-java-gc=boehm
--enable-java-awt=gtk --disable-werror i486-linux
Thread model: posix
gcc version 3.4.4 20050314 (prerelease) (Debian 3.4.3-13)
```

5. Erzeugen der fli4l Archive/Bootmedien

Sind alle Konfigurationsarbeiten erledigt, können die fli4l Archive/Bootmedien, sei es eine Bootdiskette, ein bootfähiges ISO-Image oder nur die zum Remote-Update benötigten Dateien, erstellt werden.

5.1. Erzeugen der fli4l Archive/Bootmedien unter Linux bzw. anderen Unix-Derivaten

Dies geschieht mit Hilfe von Scripts (`.sh`), die im fli4l Wurzelverzeichnis zu finden sind.

`mkfli4l.sh`

Das Build-Script erkennt selbständig die unterschiedlichen Bootvarianten (Seite 25).
Der einfachste Aufruf sieht unter Linux so aus:

```
sh mkfli4l.sh
```

Die Aktionen des Build-Scripts werden durch drei Mechanismen gesteuert:

- Konfigurationsvariable `BOOT_TYPE` aus der `<config>/base.txt`
- Konfigurationsdatei `<config>/mkfli4l.conf`
- Parameter des Build-Scripts

An Hand der Konfigurationsvariable `BOOT_TYPE` (Seite 25) entscheidet sich, welche Aktion des Build-Scripts ausgeführt wird:

- Erstellen einer fli4l Bootdiskette oder der Dual-Variante
- Erstellen eines bootfähigen fli4l CD-ISO-Images
- Bereitstellen der fli4l Dateien, zwecks Remote-Update
- usw.

Die Beschreibung der Variablen der Konfigurationsdatei `<config>/mkfli4l.conf` finden sie im Kapitel Steuerungsdatei mkfli4l.conf (Seite 271).

5.1.1. Kommandozeilenoptionen

Der letzte Steuerungsmechanismus ist das Anhängen von Optionsparametern an den Aufruf des Build-Script auf der Kommandozeile. Die Steuerungsmöglichkeiten entsprechen denen der Steuerungsdatei `mkfli4l.conf`. Die Angabe von Optionsparametern überschreiben die Werte aus der Steuerungsdatei. Aus Komfortgründen unterscheiden sich die Namen der Optionsparameter von den Namen der Variablen aus der Steuerungsdatei. Es existiert teilweise eine Kurz- und eine Langform:

```
Usage: mkfli4l.sh [options] [config-dir]

-c, --clean             cleanup the build-directory
-b, --build <dir>       sets build-directory to <dir> for the fli4l-files
-v, --verbose           verbose - some debug-output
-s, --safe              safe-mode for floppy-boot
-d, --drive X:          set target-drive to X: - default a:
    --filesonly         creates only fli4l-files - does not create a disk
    --no-squeeze        don't compress shell scripts
-h, --help              display this usage

config-dir              sets other config-directory - default is "config"

--hdinstallpath <dir>   install directly to usb/compact flash device
                        mounted or mountable to directory <dir>
                        device either has to be mounted and to be writable
                        for the user or it has to be mountable by the user

*** Remote-Update options
--remoteupdate          remote-update via scp, implies "--filesonly"
--remoteuser <name>     user name for remote-update - default is "fli4l"
--remotehost <host>     hostname or IP of remote machine - default
                        is HOSTNAME set in [config-dir]/base.txt
--remotepath <path>     pathname on remote maschine - default is "/boot"
--remoteport <portnr>   portnumber of the sshd on remote maschine

*** Netboot options
--tftpbootpath <path>   pathname to tftpboot directory
--tftpbootimage <name>  name of the generated bootimage file
--pxesubdir <path>      subdirectory for pxe files relative to tftpbootpath
```

Eine HD-pre-installation einer passend formatierten (FAT16/FAT32) CompactFlash im USB-Cardreader oder eines USB-Sticks ist über die Option `--hdinstallpath <dir>` möglich. Dieses können Sie *auf eigenes Risiko* zur Installation auf eine CompactFlash oder einen USB-Stick benutzen. Hierbei wird auf die angegebene Partition die nötigen Dateien des fli4l kopiert. Sie rufen dazu zunächst im fli4l-Verzeichnis

```
sh mkfli4l.sh --hdinstallpath <dir>
```

auf. Dabei werden die fli4l Dateien auf eine CF-Card oder USB-Stick kopiert.

Um die nächsten Schritte ausführen zu können, sind folgende Voraussetzungen zu erfüllen:

- `chmod 777 /dev/brain`
- superuser-Rechte
- installiertes `syslinux`
- installiertes `fdisk`

Durch das Script erfolgt eine Kontrolle, ob dieser Datenträger tatsächlich ein USB-Laufwerk ist und die erste Partition eine FAT-Partition ist. Anschliessend werden der Bootloader und die nötigen Dateien auf den angegebenen Datenträger kopiert. Sie erhalten eine Meldung über den Erfolg oder Misserfolg.

Nach dem Build müssen Sie

```
syslinux --mbr /dev/brain

  # make partition bootable using fdisk
  #     p - print partitions
  #     a - toggle bootable flag, specify number of fli4l partition
  #         usually '1'
  #     w - write changes and quit
  fdisk /dev/brain

  # install boot loader
  syslinux -i /dev/brain
```

ausführen. Dann sollte die CF bzw. der USB-Stick bootfähig sein. Vergessen Sie nicht den Datenträger zu unmounten.

Als letzter Optionsparameter kann ein alternatives Konfigurationverzeichnis übergeben werden. Das normale Konfigurationsverzeichnis heißt `config` und liegt direkt im fli4l Wurzelverzeichnis. An diesem Ort legen alle fli4l Pakete die Konfirgurationsdateien ab. Möchte man mehr als eine Konfiguration verwalten, so erstellt man sich ein weiteres Verzeichnis, z.B. `hd.conf`, legt dort eine Kopie der Konfigurationsdateien ab und verändert diese den Anforderungen entsprechend. Hier einige Beispiele:

```
sh mkfli4l.sh --filesonly hd.conf
sh mkfli4l.sh --no-squeeze config.test
```

5.2. Erzeugen der fli4l Archive/Bootmedien unter Windows

Es wird das Tool 'AutoIt3' verwendet (http://www.autoitscript.com/site/autoit/). Dieses ermöglicht eine 'grafische' Ausgabe, sowie Dialoge, mit denen die in den folgenden Abschnitten beschriebenen Variablen beinflusst werden können.

`mkfli4l.bat`

Das Build-Programm erkennt selbständig die unterschiedlichen Bootvarianten (Seite 25).
Der Aufruf von 'mkfli4l.bat' kann direkt aus dem Windows Explorer erfolgen, wenn man keine optionalen Parameter verwenden möchte.
Die Aktionen des Build-Programms werden durch verschiedene Mechanismen gesteuert:

- Konfigurationsvariable `BOOT_TYPE` aus der `<config>/base.txt`
- Konfigurationsdatei `<config>/mkfli4l.conf`
- Parameter des Build-Programmes
- Interaktive Einstellung in der GUI

An Hand der Konfigurationsvariable `BOOT_TYPE` (Seite 25) entscheidet sich, welche Aktion das Build-Programm ausführt:

- Erstellen einer fli4l Bootdiskette oder der Dual-Variante
- Erstellen eines bootfähigen fli4l CD-ISO-Images
- Bereitstellen der fli4l Dateien, zwecks Remote-Update
- Erzeugen der fli4l Dateien und direktes Remote-Update per SCP
- HD-pre-install einer passend formatierten CF im Cardreader
- usw.

Die Beschreibung der Variablen der Konfigurationsdatei `<config>/mkfli4l.conf` finden sie im Kapitel Steuerungsdatei mkfli4l.conf (Seite 271).

5.2.1. Kommandozeilenoptionen

Ein weiterer Steuerungsmechanismus ist das Anhängen von Optionsparametern an den Aufruf des Build-Programms auf der Kommandozeile. Die Steuerungsmöglichkeiten entsprechen denen der Steuerungsdatei `mkfli4l.conf`. Die Angabe von Optionsparametern überschreiben die Werte aus der Steuerungsdatei. Aus Komfortgründen unterscheiden sich die Namen der Optionsparameter von den Namen der Variablen aus der Steuerungsdatei. Es existiert teilweise eine Kurz- und eine Langform:

```
Usage: mkfli4l.bat [options] [config-dir]

-c, --clean             cleanup the build-directory
-b, --build <dir>       sets build-directory to <dir> for the fli4l-files
```

```
-v, --verbose           verbose - some debug-output
-s, --safe              safe-mode for floppy-boot
-d, --drive X:          set target-drive to X: - default a:
    --filesonly         creates only fli4l-files - does not create a disk
    --no-squeeze        don't compress shell scripts
-h, --help              display this usage

config-dir              sets other config-directory - default is "config"

*** Remote-Update options
--remoteupdate          remote-update via scp, implies "--filesonly"
--remoteuser <name>     user name for remote-update - default is "fli4l"
--remotehost <host>     hostname or IP of remote machine - default
                        is HOSTNAME set in [config-dir]/base.txt
--remotepath <path>     pathname on remote maschine - default is "/boot"
--remoteport <portnr>   portnumber of the sshd on remote maschine

*** GUI-Options
--no-gui                disable the config-GUI
--lang                  change language
                        [deutsch|english|espanol|french|magyar|nederlands]
```

Als letzter Optionsparameter kann ein alternatives Konfigurationverzeichnis übergeben werden. Das normale Konfigurationsverzeichnis heißt `config` und liegt direkt im fli4l Wurzelverzeichnis. An diesem Ort legen alle fli4l Pakete die Konfirgurationsdateien ab. Möchte man mehr als eine Konfiguration verwalten, so erstellt man sich ein weiteres Verzeichnis, z.B. `hd.conf`, legt dort eine Kopie der Konfigurationsdateien ab und verändert diese den Anforderungen entsprechend. Hier einige Beispiele:

```
mkfli4l.bat -s fd.conf
mkfli4l.bat -v -d b:
mkfli4l.bat --no-gui config.hd
```

5.2.2. Konfigurationsdialog – Einstellung des Konfigurationsverzeichnis

Im Hauptfenster wird die Einstellung des Konfigurationsverzeichnis angezeigt und kann ein Fenster geöffnet werden zur Auswahl des Konfigurationsverzeichnis.

Zu beachten ist, dass eine Änderung des 'Config-Dir' alle Optionen auf die Werte setzt, die in der dortigen Steuerungsdatei 'mkfli4l.conf' (Seite 271) gesetzt wurden, bzw. als Kommandozeilenparameter übergeben wurden.

Findet mkfli4l.bat kein Verzeichnis fli4l-x.y.z\config oder in dem Verzeichnis keine Datei mit dem Namen 'base.txt' öffnet sich sofort das Fenster zur Auswahl des Konfigurationsverzeichnis. Dieses ermöglicht es auf einfache Weise im fli4l-Verzeichnis mehrere Konfigurationen

zu verwalten.

Beispiel:

```
fli4l-x.y.z\config
fli4l-x.y.z\config.fd
fli4l-x.y.z\config.cd
fli4l-x.y.z\config.hd
fli4l-x.y.z\config.hd-erstellen
```

5.2.3. Konfigurationsdialog – allgemeine Einstellungen

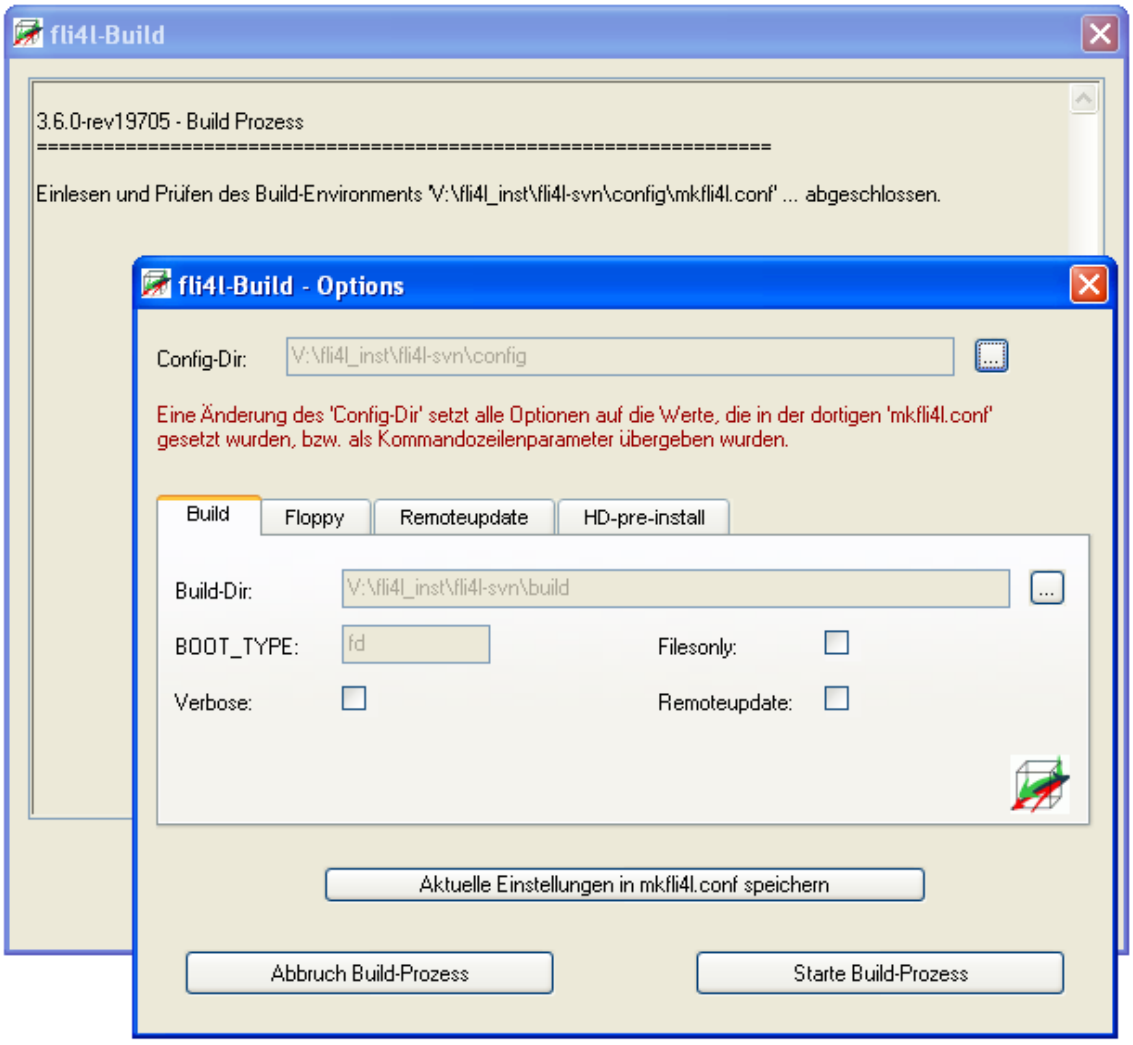


Abbildung 5.1.: Einstellungen

In diesem Dialog werden die Einstellungen für die Archiv/Diskettenerstellung festgelegt:

- Build-Dir – Verzeichnis für die Archive/CD-Images/...

- `BOOT_TYPE` – Anzeige des verwendeten/eingestellen `BOOT_TYPE` – nicht änderbar
- Verbose – Aktivierung von zusätzlichen Ausgaben während der Erstellung
- Filesonly – es werden nur die Archive erstellt – keine Diskette/kein Image
- Remoteupdate – Aktivierung des Remoteupdates per SCP

Mit der Schaltfläche **Aktuelle Einstellungen in mkfli4l.conf speichern** können die aktuell eingestellten Werte in der mkfli4l.conf gespeichert werden.

5.2.4. Konfigurationsdialog – Einstellungen für Diskette

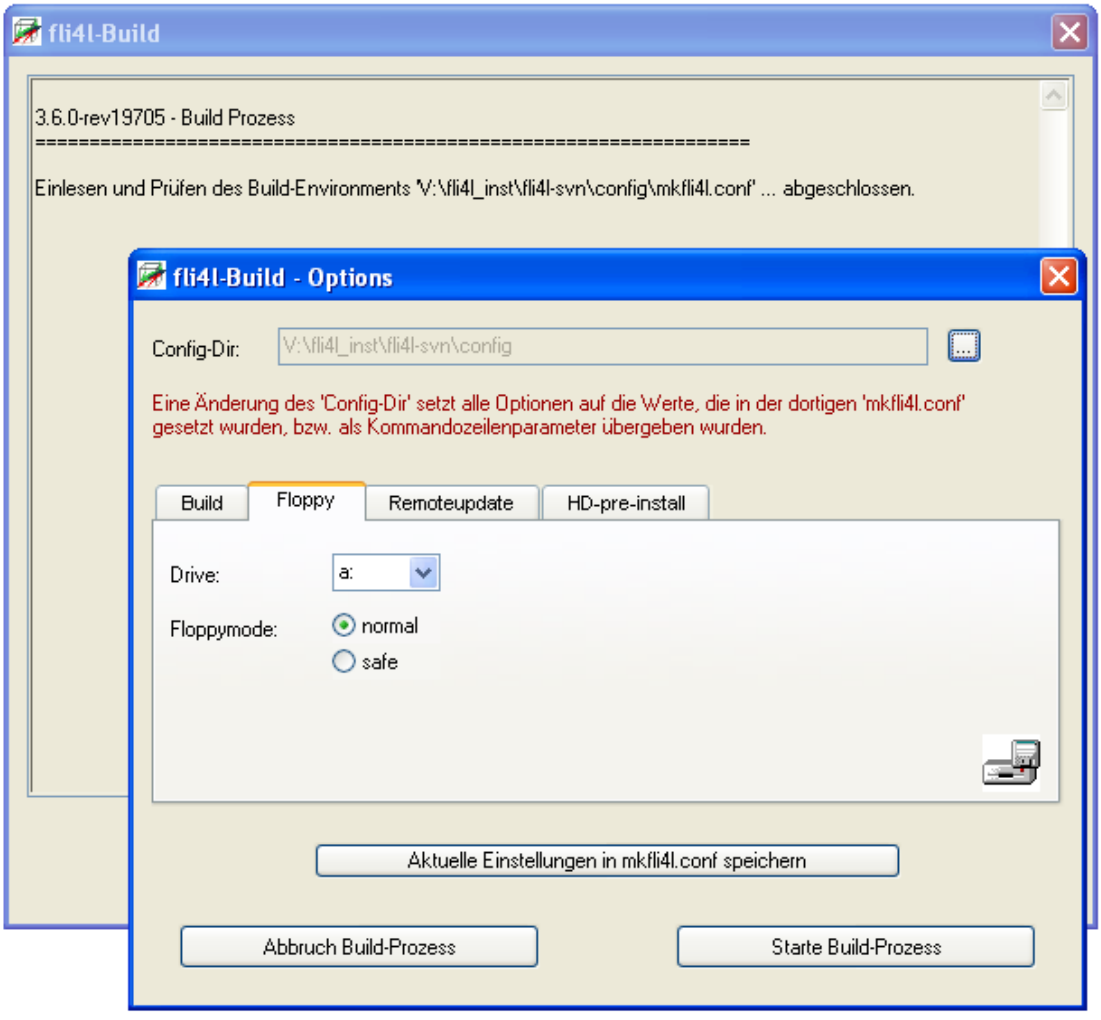


Abbildung 5.2.: Einstellungen für Diskette

In diesem Dialog werden die Einstellungen für die Erstellung einer Diskette festgelegt:

- Laufwerksbuchstabe des Diskettenlaufwerkes

- Modus für syslinux

5.2.5. Konfigurationsdialog – Einstellungen für Remoteupdate

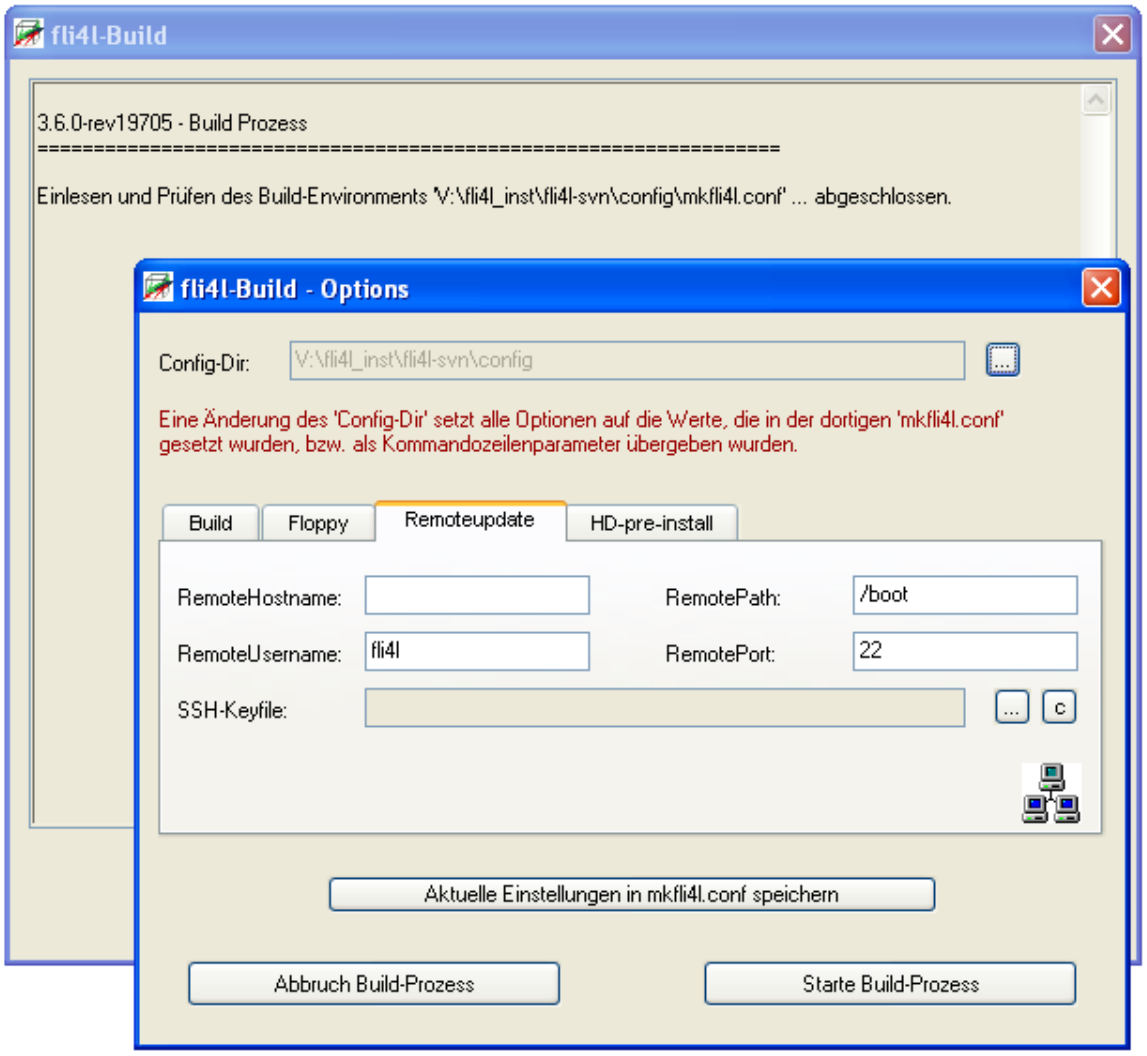


Abbildung 5.3.: Einstellungen für Remoteupdate

In diesem Dialog werden die Einstellungen für den Remoteupdate festgelegt:

- IP-Adresse oder Hostname
- Username auf dem Remotehost
- Remotepath (default:/boot)
- Remoteport (default:22)
- zu verwendendes SSH-Keyfile (ppk-Format von Putty)

5.2.6. Konfigurationsdialog – Einstellungen für HD-pre-install

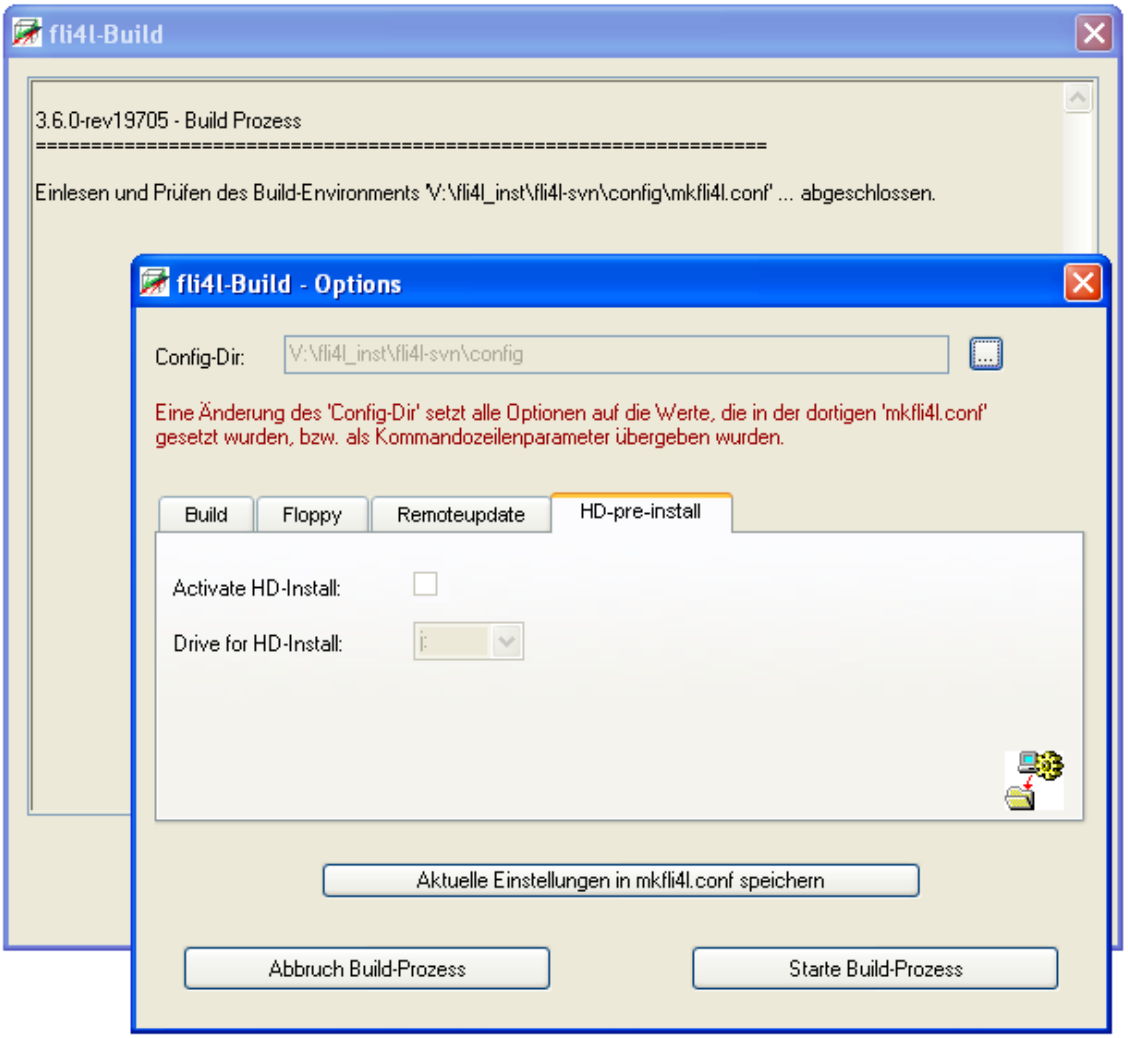


Abbildung 5.4.: Einstellungen für HD-pre-install

In diesem Dialog können die Optionen für den HD-pre-install auf einer entsprechend partitionierten und formatierten CompactFlash-Karte in einem USB-Reader eingestellt werden.
Mögliche Optionen:

- HD-pre-install aktivieren
- Laufwerksbuchstabe der CF-Karte

Hinweis zur Partionierung und Formatierung der CF: Für eine HD-Installation nach TYP A (siehe dazu Paket HD) muss auf der CF eine primäre aktive und formatierte FAT-Partitionen vorhanden sein. Möchte man weiterhin auch eine Datenpartiton benutzen, wird zusätzlich eine Linux-Partition, die mit dem Dateisystem ext3 formatiert ist, sowie die Datei `hd.cfg` auf der FAT-Partiton benötigt (hierzu sollten unbedingt die Hinweise im Paket HD beachtet werden).

5.3. Steuerungsdatei mkfli4l.conf

Seit fli4l-Version 2.1.9 existiert die Steuerungsdatei <config>/mkfli4l.conf. Durch sie werden z.B. vom Standard abweichende Verzeichnisse übergeben. Die Steuerungsdatei hat einen ähnlichen Aufbau wie die normalen fli4l Konfigurationsdateien. Alle Konfigurationsvariablen sind hier optional, d.h. sie müssen nicht in der Konfigurationsdatei vorkommen oder können als Kommentar gekennzeichnet werden.

BUILDDIR Standardwert: 'build'

Legt fest, in welchem Verzeichnis die fli4l Dateien erzeugt werden sollen. Ist die Variable undefiniert, setzt mkfli4l unter windows 'build' relativ zum fli4l Wurzelverzeichnis ein und meint damit also das Verzeichnis `build` im fli4l Wurzelverzeichnis:

```
Pfad/fli4l-x.y.z/build
```

Unter *nix setzt mkfli4l <config>/build ein und legt damit die generierten Dateien zusammen mit der Konfiguration ab.

VERBOSE Standardwert: `VERBOSE='no'`

Mögliche Werte `'yes'` oder `'no'`. Steuert die *Geschwätzigkeit* des Build Prozesses.

DRIVE Standardwert: `DRIVE='a:'`

Legt den zu verwendenden Buchstaben des Diskettenlaufwerkes fest. Unter Unix/Linux gilt aufgrund der Verwendung der MTOOLS folgende Festlegung:

- `'a:'` entspricht `/dev/fd0`
- `'b:'` entspricht `/dev/fd1`

FLOPPYMODE Standardwert: `FLOPPYMODE='normal'`

Mögliche Werte `'normal'` oder `'safe'`. Hiermit wird festgelegt, mit welchem Algorithmus `syslinux` die Bootdiskette beim Starten des Routers liest. `'safe'` ist nur bei wenigen Rechnern nötig und verzögert bei vielen Rechnern den Bootvorgang unnötig.

FILESONLY Standardwert: `FILESONLY='no'`

Mögliche Werte `'yes'` oder `'no'`. Hiermit kann das Erstellen einer Diskette abgeschaltet werden, es werden also nur die Dateien erzeugt – sinnvoll um zum Beispiel die Diskette remote updaten zu können.

REMOTEUPDATE Standardwert: `REMOTEUPDATE='no'`

Mögliche Werte `'yes'` oder `'no'`. Aktiviert das automatische Übertragen der erstellten Dateien mittels SCP auf den Router. Dieses setzt ein installiertes Paket SSHD (Seite 234) mit aktiviertem `scp` voraus. Siehe dazu auch die folgenden Variablen.

REMOTEHOSTNAME Standardwert: `REMOTEHOSTNAME="`

Gibt den Ziel-Hostnamen für den SCP Datentransfer an. Sollte kein Name angegeben sein, wird dieser der Variable `HOSTNAME` (Seite 25) entnommen.

REMOTEUSERNAME Standardwert: `REMOTEUSERNAME='fli4l'`

Username für den SCP Datentransfer.

REMOTEPATHNAME Standardwert: `REMOTEPATHNAME='/boot'`

Ziel-Pfad für den SCP Datentransfer.

REMOTEPORT Standardwert: `REMOTEPORT='22'`

Zielport für den SCP Datentransfer.

SSHKEYFILE Standardwert: `SSHKEYFILE=''`

Hier kann man eine SSH-Keydatei für den SCP-Remoteupdate angeben. Es kann somit ein Update ohne Angabe eines Passwortes erfolgen.

TFTPBOOTPATH Pfad an dem das Netboot-Image abgelegt wird.

TFTPBOOTIMAGE Name des Netboot-Images.

PXESUBDIR Unterverzeichnis für die PXE-Dateien relativ zu TFTPBOOTPATH.

HOSTS_GLOBAL Pfad und Dateiname zu einer zusätzlichen globalen Host-Datei.

SQUEEZE_SCRIPTS Aktiviert bzw. deaktiviert das Squeezen (Kommprimieren) von Scripten. Eine Script das mit Squeeze komprimiert wurde enthält z.b. keine Kommentare mehr, alle Zeileneinrückungen wurden entfernt. Im Normalfall sollte hier immer der defaultwert `'yes'` benutzt werden.

MKFLI4L_DEBUG_OPTION Es können zum Debuggen extra Optionen an das mkfli4l-Programm (Seite 301) übergeben werden.

5.4. Erstellung der Bootdiskette

Es wird zunächst ein neuer Boot-Loader `ldlinux.sys` auf der Diskette installiert. Danach werden die Dateien `syslinux.cfg` (Boot-Loader Konfiguration), `kernel` (Das eigentliche Linux), `rootfs.img` (Das Ur-Dateisystem von fli4l), `rc.cfg` (Die fli4l Konfigurationsdatei) sowie `opt.img` (Zusätzliche Programme) kopiert. In früheren Versionen reichte es, bei einer Konfigurationsänderung nur die Dateien `rc.cfg` und `opt.img` zu kopieren. Das ist bei aktuellen Versionen nicht mehr der Fall, hier wird sogar der Kernel in Abhängigkeit von der Konfiguration ausgewählt. Man sollte also sicherheitshalber immer alles kopieren.

Die Diskette(n) wird/werden dabei nicht formatiert, somit ist es absolut notwendig, eine frisch formatierte Diskette zu verwenden. Die Verwendung einer bereits teilgefüllten Diskette führt fast unweigerlich zu Platzproblemen. Eine bereits vorher für fli4l verwendete Diskette kann weiterhin benutzt werden, da hier die *störenden* Dateinamen ja die selben sind und somit einfach überschrieben werden. Wird die Diskette unter Windows erzeugt, erfolgt seit der fli4l-Version 2.1.10 eine Prüfung auf 'alte' Dateien, diese können per Bestätigung im Dialog gelöscht werden. Zusätzlich erfolgt auch vor dem Kopieren der Dateien auf die Diskette eine Prüfung des Diskettenplatzes.

Es ist auch möglich, dichter formatierte Disketten zu verwenden. Diese kann man unter Linux so formatieren:

```
fdformat /dev/fd0u1680
mformat -t 80 -h 2 -n 21 a:
```

Es stehen dann 240 KiB mehr auf der Diskette zur Verfügung.

Für Windows kann man sich entsprechende Formatierprogramme besorgen, z.B.
ftp://ftp.simtel.net/pub/simtelnet/msdos/diskutil/fdfrm18.zip
ftp://ftp.simtel.net/pub/simtelnet/msdos/diskutil/hdcp20ad.zip
http://www.moenk.de/index.php?serendipity[subpage]=downloadmanager&thiscat=2&file=1
(vgacp625.zip)
http://www.winimage.com/ (Shareware)

Die Diskette ist mit 80 Tracks, beidseitig und 21 Sectors/Track zu formatieren.

Oder man formatiert die Diskette unter fli4l. Hierzu wird eine minmale Konfiguration mit der Basis und dem Paket TOOLS erstellt. Wie üblich die Pakete entpacken, dann einen neuen Ordner `config_mtools` erstellen. In diesen Ordner die originalen Konfigurationsdateien `base.txt` und `tools.txt` aus dem Verzeicheinis `config` kopieren. Die Dateien bitte nicht verändern! Zusätzlich im Ordner `config_mtools` die Datei `_fli4l.txt` (der Unterstrich am Anfang ist wichtig!) mit folgendem Inhalt anlegen:

```
## Datei config_mtools/_fli4l.txt ##
IGNORE_IP_NET_WARNING='yes'
MOUNT_BOOT='no'
COMP_TYPE_KERNEL='lzma'
COMP_TYPE_ROOTFS='lzma'
COMP_TYPE_OPT='lzma'
NET_DRV_N='0'
IP_NET_N='0'
    PF_INPUT_N='0'
    PF_FORWARD_N='0'
    PF_POSTROUTING_N='0'
    PF_PREROUTING_N='0'
OPT_MTOOLS='yes'
```

Die Bootdisk wird dann unter Angabe des Konfigurationsordner `config_mtools` erstellt. Diese so erzeugte Diskette enthält **keine** Netzwerktreiber! Sie dient nur zum Formatieren einer Diskette im 1680 KiB Format. Nach dem Booten und Einloggen (Passwort: fli4l) das Script `fdformat.sh` aufrufen und den Aufforderungen folgen.

In seltenen Fällen kann es vorkommen, daß vorformatierte Disketten beim Booten von fli4l über 5 Minuten benötigen. Normalerweise dauert der fli4l-Boot lediglich 1 bis 2 Minuten. Wenn es wesentlich länger dauert, liegt es wahrscheinlich an einem ungüstigen Interleave oder einem unnötig eingeschalteten safe-Mode. In diesem Fall hilft nur neu formatieren – evtl. auf dem Ziel-Rechner mit purem/nacktem DOS.

Und noch eine Information:

fli4l mountet die Diskette im Root-Filesystem Read/Write, wenn die Variable `MOUNT_BOOT='rw'` (Seite 26) eingestellt ist. Dann darf kein Schreibschutz aktiviert werden. Wird der Schreibschutz trotzdem aktiviert bricht der Startvorgang ab und der fli4l-Router arbeitet nicht korrekt.

Das war's!

6. Anbindung von PCs im LAN

Für jeden Rechner im LAN ist einzustellen:

1. IP-Adresse (siehe IP-Adresse)
2. Name des Rechners plus Wunsch-Domain-Name (siehe Rechnername und Domain)
3. Standard-Gateway (siehe Gateway)
4. IP-Adresse des DNS-Servers (siehe DNS-Server)

6.1. IP-Adresse

Die IP-Adresse muss im gleichen Netz wie die IP-Adresse des fli4l-Routers (auf Ethernet-Seite) liegen, also z.B. 192.168.6.2, wenn der fli4l die Adresse 192.168.6.1 hat. Kein Rechner darf die gleiche IP-Adresse haben, weshalb man am besten (nur) die letzte Zahl ändert. Auch ist darauf zu achten, daß man hier die gleiche IP-Adresse angibt, wie man es für diesen Rechner in der Datei config/base.txt angegeben hat.

6.2. Rechnername und Domain

Der Name des Rechners ist dann z.B. “mein-pc”, die Domain “lan.fli4l”.

Wichtig: *Die im PC eingestellte Domain muss identisch mit der gewählten Domain im fli4l-Rechner sein, wenn man den fli4l-Router als DNS-Server verwenden will. Sonst kann es im Netz erhebliche Probleme geben.*

Grund: Windows-Rechner suchen regelmäßig nach Rechnern mit dem Namen ihrer Arbeitsgruppte: WORKGROUP.meine-domain.fli4l. Ist dies nicht die in fli4l eingestellte Domain (hier: meine-domain.fli4l), wird fli4l versuchen, diese Anfrage durch Weiterleiten ins Internet zu beantworten ...

Einzutragen ist die Domain in den TCP/IP Einstellungen des Rechners.

6.2.1. Windows 2000

Für Windows 2000 findet man das unter:
Start ⇒
Einstellungen ⇒
Systemsteuerung ⇒
Netzwerk- und DFÜ-Verbindungen ⇒
LAN-Verbindung ⇒
Eigenschaften ⇒
Internetprotokoll (TCP/IP) ⇒

Eigenschaften ⇒
Erweitert... ⇒
DNS ⇒
DNS-Suffix hinzufügen ⇒

"lan.fli4l" (bzw. die eingestellte domain) eingeben (ohne ""!) ⇒OK drücken.

6.2.2. NT 4.0

Start ⇒
Einstellungen ⇒
Systemsteuerung ⇒
Netzwerk ⇒
Protokolle ⇒
TCP/IP ⇒
Eigenschaften ⇒
DNS ⇒

- Hostname eintragen (eigener Rechnername)
- Domäne eintragen (wie in config/base.txt)
- IP-Adresse vom fli4l-Router hinzufügen
- DNS-Suffix hinzufügen (Domäne hinzufügen – siehe 2 Zeilen höher)

6.2.3. Win95/98

Start ⇒
Einstellungen ⇒
Systemsteuerung ⇒
Netzwerk ⇒
Konfiguration ⇒
TCP/IP (jenes, das an die Netzwerkkarte zum Router
angebunden ist) ⇒
Eigenschaften ⇒
DNS-Konfiguration:
DNS aktivieren und bei "Domäne:" dann "lan.fli4l" eingeben (ohne ""!).

6.3. Gateway

Die Angabe des Standard-Gateways ist unbedingt erforderlich, denn ohne die Angabe der richtigen IP-Adresse an dieser Stelle funktioniert nichts. Es muß hier die IP-Adresse des fli4l-Routers (auf Ethernet-Seite) angegeben werden, also z.B. 192.168.6.4 entsprechend der IP-Adresse, die hier in der Datei config/base.txt für den fli4l-Router angegeben wurde.

Es ist falsch, den fli4l-Router als Proxy in der Windows- oder Browser- Konfiguration einzutragen – außer man setzt ein Proxy auf dem fli4l-Router ein. Im Normalfall ist fli4l kein Proxy, daher bitte *nicht* fli4l als Proxy angeben!

6.4. DNS-Server

Als IP-Adresse des DNS-Servers gibt man nicht die Adresse des Provider-DNS-Servers an, sondern die des fli4l-Routers (Ethernet), da dieser nun selbst Anfragen beantworten kann bzw. diese bei Unkenntnis ins Internet weiterleitet.

Die Punkte 1 bis 4 brauchen bei konfiguriertem DHCP-Server nicht eingetragen werden, da dann der fli4l-Router die notwendigen Daten automatisch übermittelt.

Mit der Konstruktion von fli4l als DNS-Server werden viele von den Windows-PCs ausgeführten Anfragen nicht ins Internet weitergeroutet, sondern werden direkt vom fli4l-Router beantwortet.

Sonstige Anmerkungen

Internetoptionen

Bei Verbindungen muß “keine Verbindung wählen” angegeben werden. Bei Einstellungen für lokales Netzwerk(LAN): es darf hier NICHTS angegeben werden (es sei dann es wird `OPT_Proxy` verwendet).

Dank an Alexander Levenetz für die Windows-Konfigurations-Recherche :-)

7. Client-/Server-Schnittstelle imond

7.1. imon-Server imond

imond ist ein netzwerkfähiges Server-Programm, welches bestimmte Anfragen beantwortet oder auch Kommandos zur Steuerung des Routers entgegennehmen kann.

Ausserdem steuert imond das Least-Cost-Routing. Dazu verwendet er eine Konfigurationsdatei /etc/imond.conf, welche beim Booten automatisch aus den `ISDN_CIRC_x_XXX`-Variablen der Datei config/isdn.txt und anderen über ein Shell-Script erzeugt wird.

imond läuft permanent als Daemon und horcht gleichzeitig auf TCP/IP-Port 5000 und Device /dev/isdninfo.

Folgende Kommandos sind über den TCP/IP-Port 5000 möglich:

Der TCP/IP-Port 5000 ist nur vom maskierten LAN aus erreichbar. Standardmäßig wird ein Zugriff von aussen über die Firewall-Konfiguration abgeblockt.

Imond unterstützt zwei Benutzerebenen: den User- und den Admin-Modus. Für beide Ebenen kann ein Passwort gesetzt werden mittels `IMOND_PASS` bzw. `IMOND_ADMIN_PASS`. Dadurch werden die imon-Clients von imond gezwungen, eine Password-Abfrage durchzuführen und anschließend das Password an imond zu übertragen. Solange dieses Password nicht übermittelt wurde, nimmt imond nur die beiden Kommandos “pass” und “quit” entgegen. Alle anderen werden mit einem Fehler zurückgewiesen.

Möchte man das weiter einschränken, z.B. den Zugriff nur von nur einem PC erlauben, muss die Firewall-Konfiguration angepasst werden.

Die Befehle

```
enable/disable/dialmode   dial/hangup   route   reboot/halt
```

können durch die Konfigurationsvariablen `IMOND_XXX` global ein- oder abgeschaltet werden (s. Kapitel “Konfiguration”).

Mit einem Unix/Linux-Rechner (oder einem Windows-Rechner in der DOS-Box) kann man das Ganze leicht ausprobieren:

Nach Eingabe von

```
telnet fli4l 5000        \# oder entsprechender Name des fli4l-Routers
```

kann man direkt die oben aufgeführten Kommandos eingeben und sich die Ausgabe anschauen.

Zum Beispiel bekommt man mit “help” die Hilfe angezeigt, mit “quit” wird die Verbindung zum imond abgebaut.

7.1.1. Least-Cost-Routing – Funktionsweise

imond konstruiert aus der Konfigurationsdatei /etc/imond.conf (welche wiederum beim Booten aus den Konfigurationsvariablen `ISDN_CIRC_x_TIMES` usw. erstellt wird), eine zeitabhängige

Admin-Befehle

addlink ci-index	Channel zum Circuit hinzufügen (Channel-Bundling)
adjust-time seconds	Ändert die Uhrzeit des Routers um die angegebenen Sekunden
delete filename pw	Löscht die Datei auf dem Router
hup-timeout #ci-index [value]	Anzeigen bzw. Setzen des HUP-Timeout für ISDN-Circuits
removelink ci-index	Zusätzlichen Channel wieder entfernen
reset-telmond-log-file	Löschen der telmond-Protokolldatei
reset-imond-log-file	Löschen der imond-Protokolldatei
receive filename #bytes pw	Eine Datei auf den Router übertragen. Dazu quittiert imond den Befehl mit einem ACK (0x06). Danach wird die Datei in 1024er-Blöcken übertragen, die imond auch jeweils mit einem ACK bestätigt. Als letztes übermittelt imond noch ein OK.
send filename pw	Wenn das Passwort stimmt und die Datei existiert, liefert imond ein OK #bytes. Anschliessend überträgt imond die Datei in 1024er Blöcken, die jeweils mit einem ACK (0x06) bestätigt werden müssen. Als letztes liefert imond noch ein OK.
support pw	Liefert den Status/Konfiguration vom Router
sync	Synchronisiert den Cache von gemounteten Laufwerken

Admin- oder User-Befehle

dial	Wählt den Provider an (Default-Route-Circuit)
dialmode [auto|manual|off]	Liefert bzw. setzt den Dialmode
disable	Hängt ein und setzt dialmode auf “off”
enable	Setzt dialmode auf “auto”
halt	Fährt den Router sauber herunter
hangup [#channel-id]	Hängt ein
poweroff	Fährt den Router herunter und schaltet ab
reboot	Reboot vom i4l-Router!
route [ci-index]	Setzen Default-Route auf Circuit X (0=automatisch)

User-Befehle

channels	Ausgabe Anzahl der verfügbaren ISDN-Kanäle
charge #channel-id	Ausgabe der Online-Kosten für einen Channel
chargetime #channel-id	Online-Zeit unter Berücksichtigung des Taktes
circuit [ci-index]	Ausgabe eines Circuit-Namens
circuits	Ausgabe Anzahl der Default-Route-Circuits
cpu	Liefert die Auslastung der CPU in Prozent
date	Ausgabe Datum/Uhrzeit
device ci-index	Liefert das Device des Circuits
driverid #channel-id	Ausgabe Driver-Id für Channel X
help	Ausgabe Hilfe
inout #channel-id	Ausgabe der Richtung (incoming/outgoing)
imond-log-file	Ausgabe imond-Protokolldatei
ip #channel-id	Ausgabe der IP
is-allowed command	Ausgabe, ob Befehl konfiguriert/gültig ist Mögliche Befehle: dial|dialmode|route|reboot| imond-log|telmond-log|mgetty-log
is-enabled	Ausgabe, ob dialmode auf off (0) oder auto (1)
links ci-index	Ausgabe Anzahl momentaner Channel 0, 1 oder 2, 0 heisst: Kein Channel-Bundling möglich
log-dir imond|telmond|mgetty	Liefert das Logverzeichnis
mgetty-log-file	Ausgabe mgetty-Protokolldatei
online-time #channel-id	Ausgabe Online-Zeit der akt. Verbindung in hh:mm:ss
pass [password]	Abfrage, ob Password nötig bzw. Password- Eingabe 1 Userpassword gesetzt 2 Adminpassword gesetzt 4 imond befindet sich im Admin-Modus
phone #channel-id	Ausgabe Telefonnummer/Name des “Gegners”
pppoe	Liefert die Anzahl der pppoe-Devices (also 0 oder 1)
quantity #channel-id	Liefert die übertragenen Datenmengen (in Byte)
quit	Beenden der Verbindung zu imond
rate #channel-id	Ausgabe Übertragungsraten (incoming/outgoing in B/sec)
status #channel-id	Ausgabe Status für Channel X
telmond-log-file	Ausgabe telmond-Protokolldatei
time #channel-id	Ausgabe Summe Online-Zeiten, Format hh:mm:ss
timetable [ci-index]	Ausgabe der Zeittabelle für LC-Routing
uptime	Ausgabe der Uptime des Routers in Sekunden
usage #channel-id	Ausgabe Art der Verbindung, mögliche Antworten: Fax, Voice, Net, Modem, Raw
version	Ausgabe der Protokoll- und Programm-Version

Tabelle (Time-Table). Diese umfasst eine komplette Kalenderwoche im 1-Stunden-Raster (168 Stunden = 168 Bytes). Die Tabelle setzt sich jedoch lediglich aus Circuits zusammen, für die eine Default-Route definert ist.

Mit dem imond-Kommando "timetable" kann man sich diese Tabelle anschauen.

Hier ein Beispiel:

Nehmen wir an, dass 3 Circuits definiert wurden, nämlich:

```
CIRCUIT_1_NAME='Addcom'
CIRCUIT_2_NAME='Compuserve'
CIRCUIT_3_NAME='Firma'
```

wobei lediglich die ersten beiden Circuits mit Default-Routen belegt sind, also die enstprechenden Variablen ISDN_CIRC_x_ROUTE den Wert '0.0.0.0' haben.

Wenn die dazugehörigen Variablen `ISDN_CIRC_x_TIMES` folgendermaßen aussehen:

```
ISDN_CIRC_1_TIMES='Mo-Fr:09-18:0.0388:N Mo-Fr:18-09:0.0248:Y
                   Sa-Su:00-24:0.0248:Y'

ISDN_CIRC_2_TIMES='Mo-Fr:09-18:0.019:Y Mo-Fr:18-09:0.049:N
                   Sa-Su:09-18:0.019:N Sa-Su:18-09:0.049:N'

ISDN_CIRC_3_TIMES='Mo-Fr:09-18:0.08:N Mo-Fr:18-09:0.03:N
                   Sa-Su:00-24:0.03:N'
```

dann wird daraus folgende Datei /etc/imond.conf:

```
#day   hour   device   defroute   phone         name        charge   ch-int
Mo-Fr  09-18  ippp0    no         010280192306  Addcom      0.0388    60
Mo-Fr  18-09  ippp0    yes        010280192306  Addcom      0.0248    60
Sa-Su  00-24  ippp0    yes        010280192306  Addcom      0.0248    60
Mo-Fr  09-18  ippp1    yes        019160        Compuserve  0.019    180
Mo-Fr  18-09  ippp1    no         019160        Compuserve  0.049    180
Sa-Su  09-18  ippp1    no         019160        Compuserve  0.019    180
Sa-Su  18-09  ippp1    no         019160        Compuserve  0.049    180
Mo-Fr  09-18  isdn2    no         0221xxxxxxx   Firma       0.08      90
Mo-Fr  18-09  isdn2    no         0221xxxxxxx   Firma       0.03      90
Sa-Su  00-24  isdn2    no         0221xxxxxxx   Firma       0.03      90
```

imond erstellt dann im Speicher folgende Time-Table – hier die Ausgabe über das imond-Kommando "timetable":

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
-------------------------------------------------------------------------
Su  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
Mo  2  2  2  2  2  2  2  2  2  4  4  4  4  4  4  4  4  4  2  2  2  2  2  2
Tu  2  2  2  2  2  2  2  2  2  4  4  4  4  4  4  4  4  4  2  2  2  2  2  2
We  2  2  2  2  2  2  2  2  2  4  4  4  4  4  4  4  4  4  2  2  2  2  2  2
Th  2  2  2  2  2  2  2  2  2  4  4  4  4  4  4  4  4  4  2  2  2  2  2  2
Fr  2  2  2  2  2  2  2  2  2  4  4  4  4  4  4  4  4  4  2  2  2  2  2  2
Sa  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3

No.  Name                 DefRoute  Device  Ch/Min   ChInt
```

```
 1   Addcom                  no      ippp0   0.0388     60
 2   Addcom                  yes     ippp0   0.0248     60
 3   Addcom                  yes     ippp0   0.0248     60
 4   Compuserve              yes     ippp1   0.0190    180
 5   Compuserve              no      ippp1   0.0490    180
 6   Compuserve              no      ippp1   0.0190    180
 7   Compuserve              no      ippp1   0.0490    180
 8   Firma                   no      isdn2   0.0800     90
 9   Firma                   no      isdn2   0.0300     90
10   Firma                   no      isdn2   0.0300     90
```

Für den Circuit 1 (Addcom) sind also drei Zeitbereiche (1-3) eingetragen, für Circuit 2 (Compuserve) vier Zeitbereiche (4-7) und für den letzen drei Zeitbereiche (8-10).

In der Time-Table werden jeweils die Indices ausgegeben, welche für die jeweilige Stunde gültig sind. Hier tauchen lediglich die Indices 2-4 auf, da alle anderen keine LC-Default-Routen sind.

Sieht man in der Tabelle irgendwo Nullen, gibt es Lücken in den `ISDN_CIRC_X_TIMES`-Werten. Dann existiert zu diesen Zeiten keine Default-Route, Internet-Zugang abgeknipst!

Beim Programmstart ermittelt imond zunächst den Wochentag und die aktuelle Stunde. Anschließend wird dann über die Time-Table der Index ermittelt und damit dann auch der entsprechende Circuit. Auf diesen wird dann die Default-Route gesetzt.

Bei Zustandsänderungen der Channels, z.B. Wechsel von online nach offline – jedoch spätestens nach 1 Minute – geht das Spiel von neuem los: Zeit ermitteln, Lookup in Tabelle, Default-Route-Circuit ermitteln.

Ändert sich der aktuell verwendete Circuit, z.B. montags um 18:00 Uhr, wird die alte Default-Route gelöscht, eine vielleicht bestehende Verbindung beendet (sorry...) und anschließend die Default-Route auf den neuen Circuit gesetzt. Dies kann von imond bis zu 60 Sekunden später bemerkt werden, also wird spätestens um 18:00:59 umgeschaltet.

Bei Circuits, die keine Default-Route belegen, ändert sich überhaupt nichts. Hier wird der Inhalt von `ISDN_CIRC_x_TIMES` lediglich zur Berechnung der Telefonkosten verwendet. Diese können dann relevant sein, wenn man über den Client imonc das LC-Routing temporär ausschaltet und einen Circuit manuell wählt.

Man kann sich jedoch auch die Tabellen für andere Zeitbereich-Indices (im Beispiel von 1 bis 10) anschauen, auch die der "Non-LC-Default-Route-Circuits".

Kommando:

```
timetable index
```

Beispiel:

```
telnet fli4l 5000
timetable 5
quit
```

Die Ausgabe sieht dann so aus:

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
-------------------------------------------------------------------------
Su  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Mo  5  5  5  5  5  5  5  5  5  0  0  0  0  0  0  0  0  0  5  5  5  5  5  5
```

```
Tu  5  5  5  5  5  5  5  5  5  0  0  0  0  0  0  0  0  0  5  5  5  5  5  5
We  5  5  5  5  5  5  5  5  5  0  0  0  0  0  0  0  0  0  5  5  5  5  5  5
Th  5  5  5  5  5  5  5  5  5  0  0  0  0  0  0  0  0  0  5  5  5  5  5  5
Fr  5  5  5  5  5  5  5  5  5  0  0  0  0  0  0  0  0  0  5  5  5  5  5  5
Sa  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

No.  Name                  DefRoute  Device  Ch/Min   ChInt
 5   Compuserve              no      ippp1   0.0490    180
```

Alles klar?

Mit dem imond-Kommando “route” kann das LC-Routing ein- und ausgeschaltet werden. Bei Angabe eines positiven Circuit-Indices (1...N) wird die Default-Route auf den angegebenen Circuit gelegt. Ist der Index 0, wird das LC-Routing wieder aktiviert und der Circuit automatisch ausgewählt.

7.1.2. Zur Berechnung der Onlinekosten

Das ganze Modell zur Berechnung der Onlinekosten funktioniert nur korrekt, wenn der Zeittakt für einen Circuit (Variable `ISDN_CIRC_x_CHARGEINT`) über die ganze Woche konstant ist. Dies ist im Normalfall bei Internet-Providern die Regel. Wählt man sich jedoch über die Telekom (ich meine nicht T-Online!) z.B. in sein Firmennetz ein, gilt das als ganz normales Telefongespräch. Und da wechselt ab 18:00 der Takt von 90 Sekunden auf 4 Minuten (Stand Juni 00). Deshalb ist die Definition von

```
ISDN_CIRC_3_CHARGEINT='90'
ISDN_CIRC_3_TIMES='Mo-Fr:09-18:0.08:N Mo-Fr:18-09:0.03:N Sa-Su:00-24:0.03:N'
```

eigentlich nicht ganz korrekt. Es sind zwar abends umgerechnet auf die Minute 3 Pfennig (4 Minuten kosten 12 Telekom-Pfennige), jedoch ist der Takt falsch. Deshalb können bei der Kostenanzeige Differenzen zu den tatsächlichen Zahlen auftreten.

Hier ist ein Tipp, wie verschieden lange Taktzeiten dennoch richtig berücksichtigt werden können (auch wichtig für `ISDN_CIRC_x_CHARGEINT`): Man definiert einfach 2 Circuits, einen für tagsüber mit `ISDN_CIRC_1_CHARGEINT`='90' und den anderen mit `ISDN_CIRC_2_CHARGEINT`='240'. Natürlich muss man dann auch noch `ISDN_CIRC_x_TIMES` entsprechend wählen, damit tagsüber Circuit 1 und abends Circuit 2 verwendet wird.

Wie gesagt: Bei Nutzung von Verbindungen zu Internet-Providern gibt es das Problem nicht, weil dort der Zeittakt immer konstant ist und lediglich die Kosten pro Minute wechseln (oder gibt es sowas doch??? Ich traue T-* alles zu :-).

7.2. Windows-Client imonc.exe

7.2.1. Einleitung

Das Gespann imond auf dem Router und imonc auf dem Client beherrschen einen zwei Benutzermodi: den User- und den Adminmodus. Im Adminmodus sind alle Steuerelemente aktiviert. Im Usermodus steuern die Variablen `IMOND_ENABLE` (Seite 80), `IMOND_DIAL` (Seite 80), `IMOND_ROUTE` (Seite 80) und `IMOND_REBOOT` (Seite 80) ob die jeweiligen Funktionen im Usermodus zur Verfügung stehen. Sind alle diese Variablen auf ‘no’ gesetzt, bedeutet dies für die Überblick-Seite, dass alle Buttons bis auf den Exit- und den Admin-Mode-Button deaktiviert

sind. Die Entscheidung, ob der User- oder Admin-Modus benutzt wird, wird anhand des übermittelten Passwortes getroffen. Über den Button Admin-Mode, der sich in der Statusleiste befindet, kann jederzeit unter Eingabe des Admin-Passwortes vom User- zum Admin-Modus gewechselt werden. Um wieder zurück zu wechseln, muss imonc beendet und neu gestartet werden.

Sobald imonc gestartet ist, wird ein zusätzliches Tray-Icon angezeigt, welches den Verbindgungsstatus der vorhandenen Kanäle anzeigt.

Die Farben bedeuten:

Rot : Offline

Gelb : Es wird gerade eine Verbindung aufgebaut

Hellgrün : Online und Traffic auf dem Kanal

Dunkelgrün : Online und so gut wie kein Traffic auf dem Kanal

Ein etwas vom Windows-Standard abweichendes Verhalten zeigt imonc, wenn der Minimieren-Button in der Titelleiste angeclickt wird. Daraufhin minimiert sich imonc in den Systemtray und es bleibt nur noch das Tray-Icon neben der Uhr übrig. Ein Doppelklick mit der linken Maustaste auf das Tray-Symbol holt das imonc-Fenster wieder in den Vordergrund. Mit der rechten Maustaste besteht auch die Möglichkeit über das Kontextmenü, die wichtigsten imonc-Kommandos direkt auszuwählen, ohne imonc wieder auf den Bildschirm zu holen.

Viele Eigenschaften (darunter auch alle Spaltenbreiten der StringGrids) speichert imonc in der Registry, damit imonc so an die eigenen Bedürfnisse angepasst werden kann. Imonc speichert die Informationen in dem Registry-Schlüssel HKCU\Software\fli4l.

Bestehen trotz sorgfältigen Lesens der Dokumentation noch Probleme in Bezug auf imonc oder auch des Routers selber, die man z.B. in der Newsgroup posten möchte, ist es sinnvoll, auf der Über-Seite des imonc den Punkt SystemInfo auszuwählen und dort den Punkt Support Infos. Daraufhin wird das Router-Passwort abgefragt (nicht das imond-Passwort!). Imonc erstellt dann eine Datei fli4lsup.txt, welche alle wichtigen Informationen bezüglich des Routers und imonc beinhaltet. Diese Datei kann auf explizite Nachfrage in die Newsgroup gepostet werden, so daß deutlich bessere Chancen auf rasche Hilfe bestehen.

Nähere Details betreffend der Entwicklung des Windows-Clients imonc findet man auf der Homepage vom Windows ImonC-Seiten http://www.imonc.de/. Hier kann man sehen, welche neuen Features und Bug-Fixes in der nächsten Version von imonc enthalten sein werden. Ausserdem gibt es dort den neusten imonc, wenn dieser nicht schon in der fli4l-Distribution enthalten ist.

7.2.2. Startparameter

ImonC benötigt den Namen oder die IP-Adresse des fli4l-Routers. Standardmäßig versucht das Programm, eine Verbindung mit dem Rechner “fli4l” herzustellen. Wenn dieser im DNS korrekt eingetragen ist, sollte es also direkt funktionieren. Ansonsten kann man in der Verknüpfung folgende Parameter übergeben:

- /Server:IP oder Hostname des Routers (Kurzform: /S:IP oder Hostname)
- /Password:Passwort (Kurzform: /P:Password)

- /log Die Logging-Option zum Protokollieren der Kommunikation zwischen imonc und imond. Ist diese Option eingeschaltet, wird beim Beenden von imonc eine Datei imonc.log geschrieben. Diese Datei beinhaltet die gesamte Kommunikation zwischen Router und Client und wird darum sehr groß. Deshalb sollte dieser Startparameter nur gesetzt werden, wenn Probleme bestehen.
- /iport:Portnummer Die Portnummer auf die imond lauscht. Default: 5000
- /tport:Portnummer Port auf dem telmond lauscht. Default: 5001
- /rc:"Command" Das hier angegebene Kommando wird ohne weitere Überprüfung an den Router übertragen und anschliessend imonc beendet. Sollen mehrere Kommandos gleichzeitig ausgeführt werden, müssen diese durch Semikolons getrennt werden. Damit es funktioniert, muss ein gesetztes imond-Passwort mit übergeben werden, da keine Abfrage des Passwortes erfolgt. Die möglichen Kommandos sind beim imond dokumentiert, siehe Kapitel 8.1. Zusätzlich zu den dort aufgeführten Befehlen gibt es noch den Befehl timesync. Dieser bewirkt, dass die Uhrzeit des Clients mit der des Routers synchronisiert wird. Der Befehl dialtimesync wird nicht mehr unterstützt da er sich als „dial; timesync“ schreiben lässt.
- /d:"fli4l-Directory" Hiermit kann das fli4l-Directory per Startparameter übergeben werden. Interessant wenn man mit mehreren fli4l-Versionen herumspielt
- /wait Wenn der Hostname nicht aufgelöst werden kann, beendet sich imonc nicht mehr – erneuter Verbindungsaufbau durch Doppelclick auf das TrayIcon
- /nostartcheck Dieser schaltet die Überprüfung ab, ob imonc bereits läuft. Nur sinnvoll, wenn mehrere, unterschiedliche fli4l-Router in einem Netz überwacht werden sollen. Bei weiteren Instanzen werden die eingebauten Syslog- und E-Mail-Funktionalitäten deaktiviert.

Usage (einzutragen in der Verknüpfung):

```
X:\...imonc.exe [/Server:Host] [/Password:Passwort] [/iport:Portnummer]
            [/log] [/tport:Portnummer] [/rc:"Command"]
```

Beispiel mit IP-Adresse:

```
C:\wintools\imonc /Server:192.168.6.4
```

oder mit Namen und Password:

```
C:\wintools\imonc /S:fli4l /P:geheim
```

oder mit Namen, Password und Routerkommando:

```
C:\wintools\imonc /S:fli4l /P:geheim /rc:"dialmode manual"
```

7.2.3. Seite Überblick

Der Windows-Client fragt einige imond-Informationen über die bestehenden Verbindungen und bereitet sie im Anzeigefenster auf. Neben generellen Statusinformationen wie Uptime des Router oder auch der Uhrzeit sowohl lokal wie auch vom Router selber, werden für jede bestehende Verbindung die folgenden Informationen angezeigt:

Status	Verbindungsaufbau/Online/Offline
Name	Telefonnummer des Gegners oder Circuit-Name
Richtung	Zeigt an, ob es sich um eine eingehende oder ausgehende Verbindung handelt
IP	Die IP, die man zugewiesen bekommen hat
IBytes	Empfangene Bytes
OBytes	Gesendete Bytes
Online-Zeit	Aktuelle Online-Zeit
Zeit	Summe aller Online-Zeiten
KZeit	Summe Online-Zeiten unter Berücksichtigung des Zeittaktes
Kosten	Berechnete Kosten

Die Daten werden standardmäßig alle 2 Sekunden aktualisiert. Im Kontextmenü dieser Übersicht besteht die Möglichkeit für jeden vorhandenen Kanal, mit dem der Router gerade online ist, sowohl die zugewiesene IP in die Zwischenablage zu kopieren, als auch den Kanal gezielt auflegen zu können. Letzteres ist für den Fall interessant, dass mehrere unterschiedliche Verbindungen bestehen, z.B. eine um im Internet zu surfen und eine andere zur Firma, und gezielt eine dieser Verbindungen getrennt werden soll.

Ist zusätzlich auf dem fli4l-Router der telmond-Prozess aktiv, kann imonc zusätzlich Informationen über eingehende Telefonanrufe (nämlich anrufende und angerufene MSN) anzeigen. Der letzte eingegangene Telefonanruf wird oberhalb der Buttons angezeigt. Ein Protokoll der eingegangenen Telefonanrufe erhält man durch Anzeige der Seite Anrufe.

Mit den sechs Buttons im imonc können folgende Kommandos angewählt werden:

Button	Beschriftung	Funktion
1	Verbinden/Trennen	Wählen/Einhängen
2	Add link/Rem link	Kanäle bündeln: ja/nein – dieses Feature steht nur im Admin-Mode zur Verfügung
3	Reboot	fli4l neu booten!
4	PowerOff	fli4l sauber runterfahren und anschliessend ausschalten
5	Halt	fli4l sauber runterfahren, um ihn anschliessend sicher ausschalten zu können
6	Beenden	Client beenden

Die ersten fünf Kommandos können in der Konfigurationsdatei des fli4l-Routers config/base.txt für den User-Modus einzeln ein- und ausgeschaltet werden. Im Admin-Modus sind immer alle aktiviert. Die Auswahl Dialmode steuert das Wahlverhalten des Routers:

Auto	Der Router baut automatisch eine Verbindung zum Internet auf, wenn eine Anfrage aus dem lokalen Netz eintrifft.
Manuell	Der Benutzer muss selber die Verbindung aufbauen.
Aus	Es ist weder manuell noch automatisch möglich, eine Verbindung aufzubauen. Der Dial-Button ist dann deaktiviert.

Bleibt noch anzumerken, dass fli4l standardmäßig selbständig rauswählt, wenn man mit seinem Rechner in's Internet will. Man muss also eigentlich nie den Verbinden-Button drücken ...

Es besteht auch die Möglichkeit, den Default-Route-Circuit manuell zu wechseln, also das automatische LCR-Routing ein- und auszuschalten. Dafür ist in der Windows-Version von imonc die Auswahlliste "Default Route" vorgesehen. Ausserdem kann man die Hangup-TimeOut-Zeit jetzt auch über imonc direkt konfigurieren. Dazu dient der Button Config neben der Default Route. Dort werden alle konfigurierten Circuits des Routers angezeigt. Der Wert in der Spalte Hup-timeout kann für ISDN-Circuits direkt im StringGrid editiert werden (funktioniert bis dato noch nicht für DSL).

Einen Überblick über das LCR-Routing findet man auf der Seite Admin/TimeTable. Dort sieht man, welchen Circuit imond zu welcher Zeit automatisch auswählt.

7.2.4. Config-Dialog

Der Konfigurationsbereich ist über den Button Config in der Statuszeile erreichbar. Das aufgehende Fenster ist dann in die folgenden Bereiche unterteilt:

- Der Bereich Allgemein:
  - Aktualisierungsintervall: Hier wird eingestellt, wie oft die Seite Überblick aktualisiert werden soll.
  - Zeit beim Programmstart synchronisieren: Übernimmt beim Starten des Client die Zeit und das Datum des Routers als lokale Zeit. Diese Funktion kann auch manuell mit dem Button Synchronisieren auf der Überblicks-Seite aufgerufen werden.
  - Minimiert starten: Startet das Programm direkt minimiert, man sieht nur das Icon neben der Uhr.
  - Zusammen mit Windows starten: Hier kann man angeben, ob der Client direkt beim Starten von Windows mit gestartet werden soll. In dem Feld Parameter kann man die nötigen Start-Parameter angeben.
  - News von fli4l.de abholen: Sollen die News, die auf der fli4l-Homepage in der News-Sektion angezeigt werden, auch vom imonc geholt und angezeigt werden? Die Schlagzeilen werden dann in der Statusbar angezeigt. Ausserdem wird dann eine neue Seite News angezeigt, in der die kompletten Meldungen angezeigt werden.
  - Logdatei für Verbindungen: Den Dateinamen, den man hier angeben kann, wird dazu benutzt, die Verbindungs-Liste unter diesem Namen lokal auf dem Rechner abzuspeichern.
  - TimeOut für Router zum antworten: Wie lange soll auf eine Antwort der Routers gewartet werden, bevor angenommen wird, dass die Verbindung nicht mehr besteht.
  - Sprache: Hier kann die Sprache des imoncs ausgewählt werden.

  - Router Befehle bestätigen: Ist dieses Feature aktiviert, müssen alle Router-beeinflussenden Kommandos, wie zum Beispiel Reboot, Hangup ... generell bestätigt werden.
  - Auflegen auch bei Traffic: Soll kein Hinweis erfolgen, wenn die Verbindung beendet wird und noch Traffic auf der Leitung ist.
  - Automatisch Verbindung zum Router aufbauen: Soll, wenn die Verbindung zum Router unterbrochen wird (z.B. durch einen Neustart des Routers), automatisch probiert werden, die Verbindung wieder herzustellen.
  - Fenster in System Tray minimieren: Soll imonc beim Clicken auf den Beenden-Button in der Titelleiste sich in den System-Tray neben der Uhr minimieren anstatt zu beenden.

- Der Unterbereich Proxy: Hier kann ein Proxy für die http-Anfragen des imoncs definiert werden. Dieser wird dann zur Zeit für das Holen der News benutzt.
  - Proxy-Unterstützung für Http-Anfragen aktivieren: Soll ein Proxy benutzt werden
    * Adresse: Die Adresse des Proxy-Servers
    * Port: Die Portnummer des Proxy-Server (default: 8080)

- Der Unterbereich TrayIcon: Hier können die Farben des TrayIcons neben der Uhr an die eigene Bedürfnisse angepasst werden. Weiterhin kann ausgewählt werden, dass der aktuelle Dialmode als farblicher Hintergrund des TrayIcons dargestellt wird.

- Der Bereich Anrufe: Die Position des Call Notification-Fensters wird in der Registry gespeichert, so dass man sich das Fenster an die Position schieben kann, wo man es haben möchte. Es erscheint anschliessend immer wieder an dieser Stelle.
  - Aktualisierung: Hier kann ausgewählt werden, wie imonc über neue Anrufe informiert wird. Es gibt drei verschiedene Möglichkeiten. Diese erste besteht darin, regelmäßig den telmond-Dienst auf dem Router abzufragen. Eine weitere Möglichkeit besteht in der Auswertung der Syslog-Meldungen. Diese Variante ist der ersten vorzuziehen – dazu muss natürlich der Syslog-Client des imonc aktiviert sein. Wird imonc mit einem routenden eisfair eingesetzt, bietet sich noch die Möglichkeit das Capi2Text-Paket zur Anrufsignalisierung zu benutzen.
  - Führende Null wegen Telefonanlage löschen: Telefonanlage setzen manchmal eine zusätzliche Null vor die Rufnummer des Anrufer. Diese kann mit dieser Option unterdrückt werden.
  - Eigene Vorwahl: Hier kann die eigene Vorwahl hinterlegt werden. Wann dann ein Anruf mit gleichen Vorwahl eintrifft, wird die gesendete Vorwahl ausgeblendet.
  - Telefonbuch: Hier die Datei angegeben werden, in der das lokale Telefonbuch zur Auflösung von Telefonnummer gespeichert wird. Existiert die Datei nicht, wird sie vom Programm angelegt.
  - Logdatei: Den Dateinamen, den man hier angeben kann, wird dazu benutzt, die Calls-Liste unter diesem Namen lokal auf dem Rechner zu speichern. Dieser Menüpunkt ist nur sichtbar, wenn die Config-Variable `TELMOND_LOG` auf 'yes' gesetzt ist, dieses gilt auch für die eigentliche Anruf-Liste.

  - Externes Suchprogramm benutzen: In diesem Bereich kann ein Programm angegeben werden, dass aufgerufen wird, wenn eine Telefonnummer mittels des lokalen Telefonbuches nicht aufgelöst werden kann. Nähere Infos sollten den entsprechenden Programmen beiliegen. Bis jetzt gibt es eine Anbindung an die Telefonbuch-CD KlickTel sowie von Marcel Wappler eine Anbindung an die Palm-Datenbank.

- Der Unterbereich Call Notification: Hier kann das bestimmt werden, ob ein Hinweis auf eingehende Telefonanrufe angezeigt werden soll und wie dieser sich optisch präsentiert.
  - Call Notification aktivieren: Bestimmt, ob Anrufe signalisiert werden sollen.
  - Call Notification anzeigen: Soll bei eingehenden Anrufen ein Hinweisfenster mit den Infos: angerufene MSN, Rufnummer des Anrufers und Datum/Uhrzeit erscheinen? Dafür ist es nötig, dass in der Datei config/isdn.txt die Variable `OPT_TELMOND` auf 'yes' gesetzt wird.
    * Unterdrücken, wenn keine Nummer übertragen wurde: Soll Die Call Notification nicht angezeigt werden, wenn keine Rufnummer übertragen wurde.
    * Anzeigendauer: Diese Angabe beeinflußt die Dauer, wie lange das Call Notification-Fenster geöffnet bleiben soll. Die Angabe von "0" an dieser Stelle bewirkt, dass das Fenster nicht automatisch geschlossen wird.
    * Fontsize: Hiermit wird die Schriftgröße bestimmt. Dieses hat einen Einfluss auf die Größe des Fenster, da die notwendige Größe des Fenster anhand der tatsächlichen Größe der Mitteilung berechnet wird.
    * Farbe: Hiermit kann die Schriftfarbe ausgewählt werden. Ich selber benutzte die Farbe rot, damit ich es auch direkt wahrnehme.

- Der Unterbereich Phonebook: Die Seite Phonebook beinhaltet das Telefonbuch, welches zur Rufnummerauflösung der anrufenden Nummer als auch der eigenen MSN benutzt wird. Die Seite wird auch angezeigt, wenn die Konfigurationsvariable `TELMOND_LOG` auf 'no' gesetzt ist, da die Rufnummerauflösung auch für die Anzeige des letzten Anrufes auf der Summary-Seite benutzt wird. Alternativ kann statt dem Telefonbuch auf dem Router auch eine lokale Datei ausgewählt werden.

  Der Aufbau der Eintrag sieht wie folgt aus:

```
# Format:
# Telefonnummer=anzuzeigender Name[, Wavefilename]
# 0241123456789=Testuser
00=unbekannt
508402=Fax
0241606*=Elsa AG Aachen
```

  Dabei sind die ersten drei Zeilen Kommentare. Die vierte Zeile bewirkt, dass, wenn keine Rufnummer übermittelt wird, "unbekannt" angezeigt wird. In der fünften Zeile wird der MSN 508402 der Name "Fax" zugeordnet. Ansonsten ist das Format immer Telefonnummer=Name, der stattdessen angezeigt werden soll. In der sechsten Zeile ist noch die Möglichkeit demonstriert, eine Sammelrufnummer zu definieren. Damit wird erreicht, dass für alle Nebenstellen von 0241606 der Name angezeigt wird. Zu beachten hierbei ist, dass der erste Eintrag im Telefonbuch, welcher auf den Anruf passt, genommen

wird. Optional kann auch noch ein Wave-Datei angegeben werden, die abgespielt wird, wenn ein Telefonanruf von dieser Rufnummer eingeht.

Ab der Version 1.5.2 besteht jetzt auch die Möglichkeit auf der Seite Names das lokale Telefonbuch mit dem auf dem Router abgespeicherten (in /etc/phonebook) zu synchronisieren und umgekehrt. Dabei werden nicht nur einfach die Dateien ersetzt, sondern es werden die noch fehlende Einträge hinzugefügt. Gibt es eine Telefonnummer in beiden Telefonbüchern mit unterschiedlichen Namen, wird nachgefragt, welcher Eintrag genommen werden soll. Für die Synchronisierung des Telefonbuches auf dem Router ist noch anzumerken, dass dieses nur in der Ramdisk verändert wird, d.h. dass nach einem Reboot sämtliche Änderungen verloren gehen.

- Der Bereich Sound: Die Wave-Dateien, die hier angegeben werden, werden abgespielt, wenn das jeweilige Ereigniss eingetreten ist.
  - E-Mail: Wenn der E-Mail-Checker auf einem angegebenen POP3-Server neue E-Mails vorfindet, wird die angegebene Wave-Datei abgespielt.
  - E-Mail-Error: Wenn ein Fehler beim Abrufen der E-Mails auftritt, wird diese Wave-Datei abgespielt.
  - Verbindung verloren: Wenn die Verbindung zum Router nicht mehr vorhanden ist (z.B. wenn der Router von einem anderen Client gerade neu gebootet wird), wird diese Wave-Datei abgespielt. Wenn die Option “Automatic Reconnect to router” nicht aktiviert ist, erscheint ausserdem eine MessageBox, die nachfragt, ob versucht werden soll, eine neue Verbindung zum Router aufzubauen.
  - Verbindungsmeldung: Wenn der Router eine Verbindung zum Internet aufgebaut hat, wird diese Wave-Datei abgespielt.
  - Verbingsabbau: Wenn der Router die Verbindung zum Internet wieder abgebaut hat, wird diese Wave-Datei abgespielt.
  - Anrufmeldung: Wenn die Call Notification aktiviert ist und ein neuer Anruf eingeht, wird die angegebene Wave-Datei abgespielt.
  - Fax Notification: Die heier angegebene Wave-Datei wird nach dem Empfang neuer Faxe abgespielt.
- Der Bereich E-Mails
  - Accounts: Dieser Bereich dient dazu, die vorhandenen POP3-Accounts zu konfigurieren.
  - E-Mail-Check aktivieren: Soll der E-Mail-Checker automatisch nach neuen E-Mails suchen
    - Check jede x Min: Hiermit wird angegeben, wie oft der E-Mail-Checker automatisch nach neuen E-Mails suchen soll. Achtung: ein zu kurzes Intervall kann dazu führen, dass der Router komplett online bleibt! Dies ist der Fall, wenn das Intervall kleiner ist als der Hangup-Timeout des verwendeten Circuits.
    - TimeOut x Sec: Wie lange soll auf einen einen POP3-Server gewartet werden, bis er antwortet. Der Wert “0” bedeutet, dass kein TimeOut gesetzt wird.

    * Auch wenn Router offline: Hiermit wird erreicht, dass der Router sich selbstständig einwählt, um nach E-Mails zu sehen. Nachdem alle POP3-Konton nach E-Mails überprüft wordensind, wird die Verbindung wieder getrennt. Um dieses Feature nutzen zu können, muss Dialmode auf 'auto' stehen. Achtung: Dadurch entstehen zusätzliche Kosten, wenn nicht gerade eine Flatrate benutzt wird!
    * Zu benutzender Circuit: Hiermit wird angegeben, welcher Circuit zur Einwahl beim E-Mail-Checken benutzt werden soll.
    * Anschliessend online bleiben: Soll direkt nach dem E-Mail-Check direkt die Verbindung getrennt werden oder eine Verbindungstrennung durch das Hanguptimeout realisiert werden.
    * E-Mail-Header laden: Sollen auch die E-Mail-Header geladen oder nur die Anzahl der vorhandenen E-Mails abgefragt werden? Das Laden der E-Mail-Header ist Voraussetzung, wenn man E-Mails direkt auf dem S erver löschen möchte.
    * Notify only new E-Mails: Sollen nur neue E-Mails akustisch und mit dem Tray-Icon gemeldet werden
    * E-Mail-Client starten: Soll der angegebene E-Mail-Client automatisch gestartet werden, wenn neue E-Mails vorhanden sind.
    * E-Mail-Client: Hier wird der zu startende E-Mail-Client angegeben.
    * Param: Hier kann man zusätzliche Parameter angeben, die beim Start des E-Mail-Clients übergeben werden sollen. Wenn Outlook als E-Mail-Client benutzt wird (nicht Outlook Express), sollte /recyle als Parameter eingetragen werden, damit eine bereits geöffnete Instanz von Outlook beim Eintreffen von neuen E-Mails benutzt wird.

- Der Bereich Admin
  - root-Passwort: Hier sollte das Router-Passwort (in config/base.txt unter `PASSWORD` eingetragen) eingetragen werden, damit z.B. das Portforwarding lokal bearbeitet und wieder auf dem Router hinterlegt werden kann.
  - Dateien auf dem Router, die angezeigt werden sollen: Alle hier angegebenen Dateien, die sich auf dem Router befinden, können einfach per Maus-Click auf der Seite Admin/Dateien angezeigt werden. Somit kann man sich auf einfache Weise die Logfiles des Routers direkt im imonc anzeigen lassen.
  - Konfigdateien bearbeiten: Hier kann ausgewählt werden, ob die Konfigdateien alle im Editor geöffnet werden sollen (dies kann, wenn TXT-Dateien noch mit einem einfachen Editor verknüpft sind, dazu führen, dass sehr viele Instanzen des Editors geöffnet werden). Alternativ kann auch einfach nur das Verzeichnis geöffnet werden, so dass die Möglichkeit besteht, nur die Dateien auszuwählen, die bearbeitet werden sollen.
  - DynEisfaiLog: Wenn ein Account bei DynEisfair vorhanden ist, kann man hier seine Zugangsdaten eintragen und sich dann ein Log der Aktualisierung des Dienstes auf der Seite Admin/DynEisfairLog anschauen.

- Der Bereich LaunchList dient dazu, die Launchliste zu konfigurieren. Diese wird nach einem erfolgreichen Connect ausgeführt, wenn die Option "Activate Launchlist" aktiviert ist.

  - Programme: Alle hier eingetragenen Programme werden automatisch gestartet, sobald der Router eine Verbindung aufgebaut hat und die Launchliste aktiviert ist.
  - LaunchList aktivieren: Soll die Launchliste beim erfolgreichen Verbindungsaufbau ausgeführt werden?

- Der Bereich Traffic dient dazu, dass TrafficInfo-Fenster den eigenen Bedürfnissen anzupassen. Von einem User habe ich den Hinweis bekommen, dass es mit älteren DirectX-Versionen offenbar Darstellungsfehler gibt.
  - Separates Traffic-Info-Fenster anzeigen: Soll eine grafische Kanalauslastung in einem separaten Fenster angezeigt werden? In dem Kontextmenü des Fensters kann man festlegen, ob das Fenster das Attribut StayOnTop bekommen soll. Dieses bewirkt, dass sich das Fenster immer über allen anderen Fenstern plaziert. Auch dieser Wert wird in der Registry abgespeichert und steht somit auch nach einem erneuten Programmstart wieder zur Verfügung.
  - Titelleiste anzeigen: Soll die Titelleiste des Traffic-Info-Fensters angezeigt werden? In der Titelzeile wird angezeigt, mit welchem Circuit der Router gerade online ist.
    * CPU-Auslastung in Titelleiste: Soll auch die CPU-Auslastung in der Titelzeile angezeigt werden?
    * Online-Zeit in Titelleiste: Soll die Onlinezeit des Kanals auch in der Titelzeile angezeigt werden?
  - Semi-transparentes Fenster: Soll das Fenster transparent dargestellt werden? Diese Funktion steht nur unter Windows 2000 und Windows XP zur Verfügung.
  - Farben: Hier werden die Farben für das TrafficInfo-Fenster definiert. Zu Berücksichtigen ist dabei, dass der DSL-Kanal und der erste ISDN-Kanal die selben Farbwerte zugewiesen bekommen.
  - Limits: Hier können die max. Übertragungswerte für DSL eingestellt werden – für T-Online gilt: DSL Upload 128 und DSL Download 1024.

- Der Bereich Syslog dient dazu, die Anzeige der Syslog-Meldungen zu konfigurieren.
  - Syslog-Client aktivieren: Sollen Syslog-Meldungen im imonc angezeigt werden? Diese Option sollte ausgeschaltet sein, wenn ein externer Syslog-Client, wie zum Beispiel Kiwi's Syslog Client, benutzt wird.
  - Alle Meldungen ab Stufe anzeigen: Ab welcher Prioritätsstufe sollen die Syslog-Meldungen angezeigt werden? Es ist sinnvoll am Anfang mit der Stufe Debug anzufangen, um damit festzustellen, welche Meldungen einen interessieren. Anschliessend kann hier dann die entsprechende Stufe eingetragen werden.
  - Syslog-Meldungen in einer Datei speichern: Sollen die angezeigten Syslog-Meldungen in einer Datei gespeichert werden? In der Groupbox können dann die Meldungen ausgewählt werden, die in der Datei geloggt werden sollen. Für den Dateinamen sind folgende Platzhalter eingefügt worden:

    %y – wird durch das aktuelle Jahr ersetzt

    %m – wird durch den aktuellen Monat ersetzt

    %d – wird durch den aktuellen Tag ersetzt

  - Portnamen anzeigen: Sollen statt den Portnummern deren Bedeutung angezeigt werden?
  - Firewall-Meldungen auch im User-Modus anzeigen: Hiermit wird festgelegt, dass Firewall-Meldungen auch im User-Modus angezeigt werden sollen.

- Der Bereich Fax dient dazu, die Faxanzeige vom imonc zu konfigurieren. Damit dieser Punkt angezeigt wird, muss mgetty bzw. faxrcv auf dem Router installiert sein (zu finden als OPT-Pakete auf der fli4l-Homepage).
  - Logdatei für Faxe: Den Dateinamen, den man hier angeben kann, wird dazu benutzt, die Fax-Liste unter diesem Namen lokal auf dem Rechner abzuspeichern.
  - Lokales Verzeichnis: Um die Faxe anzeigen zu können, müssen sie lokal gespeichert werden. Dieses kann hier eingestellt werden.
  - Aktualisierung: Es gibt zwei verschiedene Möglichkeiten, wie imonc mitbekommt, dass ein neues Fax eingegangen ist. Entweder wertet imonc die entsprechenden Syslogmeldungen aus (dazu muss natürlich der Syslog-Client im imonc aktiviert sein) oder er schaut regelmäßig selber in der Logdatei nach. Die erste Variante ist zu bevorzugen. Falls die zweite Variante genutzt wird, kann man noch angeben, wieoft die Faxübersichtsseite aktualisiert werden soll. Dabei ist zu beachten, dass dieser Wert keine Angabe in Sekunden ist, sondern noch mit der Angabe von Allgemein/Aktualisierungsintervall multipliziert wird.

- Der Bereich Grids dient dazu die Grids (Tabellen) im imonc an die eigenen Bedürfnisse anzupassen. Einerseits kann für jedes Grid angegeben werden, welche Spalten angezeigt werden sollen, andererseits gibt es die Möglichkeit für die Grids im Bereich Anrufe, Verbindungen und Faxe anzugeben, von wann ab die Infos angezeigt werden sollen.

7.2.5. Seite Anrufe

Die Seite Calls wird nur angezeigt, wenn die Konfigurationsvariable `TELMOND_LOG` auf 'yes' eingestellt ist, denn sonst wird kein Anruf-Log geführt. Auf dieser Seite werden alle abgespeicherten Telefonanrufe angezeigt, die eingegangen sind, wie der Router eingeschaltet war. Dabei kann umgeschaltet werden zwischen der Ansicht der lokal gespeicherten Anrufe oder nur der auf dem Router gespeicherten Anrufe. Wird bei der Anzeige der auf dem Router gespeicherten Anrufe der Zurücksetzen-Button gedrückt, wird das Logfile auf dem Router gelöscht.

In der Anruf-Übersicht kann mit der rechten Maustaste auf der Rufnummer oder der eigenen MSN diese ins Telefonbuch übernommen werden, um der Rufnummer bzw. MSN dort einen Namen zuzuweisen, der dann stattdessen angezeigt wird.

7.2.6. Seite Verbindungen

Neu ist ab der Version 1.4 die Anzeige der vom Router aufgebauten Verbindungen zum Internet. Diese befindet auf der Seite Connections. Somit hat man einen guten Überblick, wie sich der Router bei der automatischen Einwahl ins Internet verhält. Damit diese Seite angezeigt werden kann, muss in der Datei config/base.txt die Variable `IMOND_LOG` auf 'YES' gesetzt werden.

Genauso wie bei der Anruf-Übersicht kann auch hier zwischen den lokal gespeicherten und auf dem Router gespeicherten Verbindungen umgeschaltet werden. In der Ansicht der auf dem

Router gespeicherten Verbindungen bewirkt ein Drücken des Zurücksetzen-Buttons, dass das Logfile auf dem Router gelöscht wird.

Angezeigt werden pro Verbindung

- Provider
- Startdatum und -zeit
- Enddatum und -zeit
- Onlinezeit
- Abrechnungszeit
- entstandene Kosten
- empfangene Zeichen
- gesendete Zeichen

7.2.7. Seite Fax

Damit die Seite Faxe angezeigt wird, muss auf dem Router entweder das `OPT_MGETTY` von Michael Heimbach oder `OPT_MGETTY` von Felix Eckhofer installiert werden. Diese gibt es auf der fli4l-Homepage unter OPT-Pakete. Auf dieser Seite werden dann alle eingegangenen Faxe aufgelistet. Das Kontextmenü der Übersicht bietet mehrere Möglichkeiten, diese stehen allerdings nur im Admin-Modus zur Verfügung:

- Es kann ein Fax angezeigt werden. Dazu muss unter Admin/Remoteupdate der Pfad für das fli4l-Verzeichnis korrekt gesetzt werden, da die Faxe auf dem Router in gepackter Form vorliegen und somit gzip aus dem fli4l-Paket benötigt wird. Alternativ kann gzip.exe und win32gnu.dll auch ins imonc-Verzeichnis kopiert werden. Kann gzip.exe nicht an einer der beiden Stellen gefunden werden, wird stattdessen der Webserver des Routers probiert zu öffnen (direkt mit dem Aufruf des richtigen CGIs).
- Ein einzelnes Fax kann gelöscht werden. Dabei wird das Fax sowohl lokal als auch auf dem Router gelöscht (sowohl die eigentliche Faxdatei, als auch der Eintrag in den Logdateien).
- Sämtliche auf dem Router befindlichen Faxe löschen. Damit werden die Faxe und die Logdatei auf dem Router gelöscht. Die Faxe werden nicht aus der lokalen Logdatei gelöscht.

Genauso wie bei der Anruf-Übersicht kann auch hier zwischen den lokal gespeicherten und auf dem Router gespeicherten Faxen umgeschaltet werden.

7.2.8. Seite E-Mail

Diese Seite wird nur angezeigt, wenn im Config-Dialog mindestens ein aktiviertes POP3-E-Mail-Konto eingerichtet worden ist.

Die Seite E-Mail dürfte sich eigentlich selber erklären. Hiermit wird der mittlerweile eingebaute E-Mail-Checker beobachtet. Ist die Option “Check even if the router is offline” nicht

aktiviert, überprüft der E-Mail-Checker alle E-Mail-Konten nach E-Mails, sobald der Router online ist und anschließend im eingestellt Intervall. Ist die genannte Option aktiviert, überprüft der E-Mail-Checker im eingestellten Intervall. Ist der Router gerade online, wird die bestehende Verbindung benutzt. Ist er nicht online, wird eine Verbindung selbständig mit dem ausgewählten Circuit hergestellt, die, sobald alle E-Mail-Konten abgearbeitet sind, wieder getrennt wird. Damit man diese Option nutzen kann, muss Dialmode auf "auto" stehen.

Sind E-Mails auf dem POP3-Server vorhanden, wird entweder automatisch der eingestellte E-Mail-Client gestartet oder ein neues Symbol im Tray neben der Uhr angezeigt, welches als Hint die Anzahl der E-Mails auf jedem Server liefert. Ein Doppelclick startet dann den eingestellten E-Mail-Client. Ist ein Fehler bei einem der E-Mail-Konten aufgetreten, erscheint einerseits ein Hinweis in der E-Mail-History, andererseits wird das E-Mail-TrayIcon angezeigt, welches dadurch gekennzeichnet ist, dass die obere rechte Ecke rot gefärbt ist.

In der E-Mail-Übersicht kann man mit dem Kontextmenü Mails direkt auf dem Server löschen, ohne sie vorher komplett downloaden zu müssen. Dies geschieht, indem mit der rechten Maustaste das Kontextmenü angezeigt wird. Dabei sollte eine Zelle der entsprechenden Zeile markiert sein, wo die zu löschende E-Mail eingetragen ist. Im Kontextmenü wählt man den einzige Punkt Delete MailMessage aus.

7.2.9. Admin

Dieser Abschnitt steht nur zur Verfügung, wenn sich imonc im Admin-Modus befindet.

Der erste Punkt lieferte eine Übersicht über die verwendeten Circuits – sprich Internetprovider – die der Router automatisch per LCR auswählt. Ein Doppelclick auf einen Provider in der Providerübersicht zeigt an, für welche Zeiträume der Circuit in config/base.txt definiert worden ist.

Der zweite Punkt dort ist die Möglichkeit ein Fernupdate auf dem Router einzuspielen. Dabei kann ausgewählt werden, welche fünf Programmpakete (Kernel, Rootfilesystem, Opt-Datei, rc.cfg und syslinux.cfg) auf den Router kopiert werden sollen. Damit man das Update einspielen kann, muss man zuerst mal das fli4l-Verzeichnis angeben, damit imonc weiss, woher es die nötigen Dateien nehmen soll. Ausserdem muss angegeben werden, in welchem Unterverzeichnis die Konfigurationsdateien liegen (standardmäßig config), um die Opt-Datei, rc.cfg und syslinux.cfg jeweils neu zu erzeugen. Es ist ratsam, einen Reboot nach dem Einspielen des Updates durchführen, damit die Änderungen auch direkt wirksam werden. Wird während des Updates nach einem Passwort nachgefragt, ist das Passwort gemeint, welches in config/base.txt unter PASSWORD eingetragen ist.

Um die Beschränkung des Port-Forwarding zu umgehen, dass ein Port nur an genau einen Client-Rechner gebunden ist, besteht jetzt die Möglichkeit, die Konfiguration auf dem Router zu editieren. Damit die Änderungen aktiv werden, muss die Verbindung neu hergestellt werden. Da die Datei nur in der Ramdisk ersetzt wird, bleiben die Änderungen nur bis zum nächsten Neustart des Routers erhalten. Um die Änderungen dauerhaft zu speichern, muss ein neues Opt-File auf dem Router installiert werden mit einer geeignet angepassten base.txt aus dem Config-Verzeichnis.

Der vierte Punkt auf der Admin-Seite – Dateien – dient dazu, Konfigurations- und Logdateien des Routers einfach per Maus-Click anzuzeigen. Die Auswahlliste wird über den Punkt Config/Admin und dort "files on router to view" konfiguriert. Anschliessend kann einfach über die ComboBox auf dieser Seite ausgewählt werden, welche Datei angezeigt werden soll.

Der fünfte Punkt ist die Seite DynEisfairLog, sie erscheint nur wenn im Config-Dialog unter

Admin die Zugangsdaten des DynEisfair-Accounts eingetragen worden sind. Ist dies geschehen, wird auf dieser Seite Log des Dienstes angezeigt.

Als letzten Punkt gibt es noch die Seite Hosts. Hier werden alle in der Datei /etc/hosts eingetragenen Rechner angezeigt. Weiterhin wird probiert jeden dieser dort eingetragenen Rechner anzupingen und das Ergebnis davon wird ebenfalls angezeigt. Somit kann man schnell rausbekommen, welche dieser Rechner eingeschaltet sind.

7.2.10. Seiten Fehler, Syslog und Firewall

Die Seiten Fehler, Syslog und Firewall werden nur angezeigt, wenn in den entsprechenden Logs Einträge vorhanden sind. Die Einträge der Seiten Syslog und Firewall werden nur angezeigt, wenn man im Admin-Modus ist.

Auf der Seite Fehler werden sämtliche imonc/imond-spezifischen Fehler festgehalten. Wenn Probleme bestehen, kann unter Umständen ein Blick auf diese Seite die Ursache der Probleme anzeigen.

Auf der Seite Syslog werden die ankommende Syslog-Meldungen angezeigt, bis auf die Meldungen der Firewall. Diese werden auf der eigenen Seite Firewall dargestellt. Damit dies funktioniert, muss die Variable `OPT_SYSLOGD` in der Konfigurationsdatei config/base.txt auf 'yes' gesetzt werden. Ausserdem muss die Variable `SYSLOGD_DEST` auf die IP des Clients gesetzt werden (genau: `SYSLOGD_DEST`='@100.100.100.100 – wobei die IP natürlich an die IP des Clients angepasst werden muss). Angezeigt wird neben der eigentlich Syslog-Meldung auch Datum, Uhrzeit, IP des Senders und die Prioritätsstufe.

Damit die Firewall-Meldungen bei den ganzen Syslog-Meldungen nicht untergehen, werden diese auf der separaten Seite Firewall angezeigt. Damit die Firewall- Meldungen angezeigt werden können, muss zusätzlich in der Datei config/base.txt die Konfigurationsvariable `OPT_KLOGD` auf 'yes' gesetzt werden.

7.2.11. Seite News

Auf dieser Seite werden, vorausgesetzt die Option ist im Config-Bereich des Imonc aktiviert, die News, welche auf der fli4l-Homepage angezeigt werden, auch direkt im Imonc angezeigt werden. Dazu wird mittels des http-Protokolls die URL http://www.fli4l.de/german/news.xml abgerufen. Neben den News werden mittlerweile auch die fünf aktuellsten Opt-Pakete angezeigt. Dazu wird die URL http://www.fli4l.de/german/imonc_opt_show.php abgefragt. Außerdem wird in der Statusleiste vom Imonc die Überschriften der News alternierend angezeigt.

7.3. Unix/Linux-Client imonc

Für Linux gibt es mittlerweile 2 Versionen: eine textbasierte (imonc) und eine mit graphischer Oberfläche (ximonc). Den Source zu ximonc findet man im Verzeichnis src. Die Dokumentation für ximonc wird erst in der 1.5-Final-Version zur Verfügung stehen. Ein erfahrener Linux-User sollte aber mit den Sources keine Probleme haben.

Beschränken wir uns daher hier zunächst auf die textbasierte Version von imonc: Dieses ist ein curses-basiertes Programm, hat also keine graphische Oberfläche. Der Source liegt im Verzeichnis unix.

Installation:

```
cd unix
make install
```

imonc wird dabei in /usr/local/bin installiert.
Aufruf:

```
imonc hostname
```

Dabei ist als hostname der Name oder die IP-Adresse des fli4l-Routers anzugeben, also z.B.

```
imonc fli4l
```

imonc zeigt folgende Informationen:

- Datum/Uhrzeit des fli4l-Routers
- Momentan eingestellte Route
- Default-Route-Circuits
- ISDN-Kanäle

  Status : Calling/Online/Offline

  Name : Telefonnummer des Gegners oder Circuit-Name

  Time : Online-Zeit

  Charge-Time : Online-Zeit unter Berücksichtigung des Zeittaktes

  Charge : Berechnete Kosten

Mögliche Kommandos sind:

Nr	Befehl	Bedeutung
0	quit	Programm beenden
1	enable	Aktivieren
2	disable	Deaktivieren
3	dial	Wählen
4	hangup	Einhängen
5	reboot	Neu booten
6	timetable	Zeittabelle ausgeben
7	dflt route	Neuen Default-Route-Circuit bestimmen
8	add channel	2. Kanal hinzuschalten
9	rem channel	2. Kanal deaktivieren

Zu den Kommandos im Einzelnen:

0 – quit Die Verbindung zum imond-Server wird abgebaut und das Programm beendet.

1 – enable Alle Cirucits werden auf dialmode “auto” gestellt. Das ist auch der Default-Zustand von fli4l nach dem Booten. Das heisst, dass fli4l bei einem Verbindungsaufbauwunsch eines Rechners im Netz automatisch rauswählt.

2 – disable Alle Circuits werden auf dialmode “off” gestellt. Damit ist fli4l so gut wie tot, bis er mit dem Enable-Kommando wieder geweckt wird.

3 – dial Manuelle Wahl auf dem Default-Route-Circuit. Ist eher für Testzwecke gedacht, da fli4l normalerweise automatisch wählt.

4 – hangup Manuelles Einhängen: damit kann man dem automatischen Einhängen von fli4l zuvorkommen.

5 – reboot fli4l wird neu gebootet. Ziemlich überflüssiges Kommando ...

6 – timetable Es wird die Zeittabelle für die Default-Route-Circuits ausgegeben. Beispiel: s.o.

7 – default route circuit Manuelles Wechseln des Default-Route-Circuits. Kann z.B. dann sinnvoll sein, wenn man das automatische LC-Routing von fli4l für eine Weile ausser Kraft setzen will, da einige Provider einen Zugriff auf das eigene Postfach nur über den eigenen Internet-Zugang erlauben.

8 – add channel Hier kann der 2. ISDN-Kanal manuell hinzugeschaltet werden. Voraussetzung: `ISDN_CIRC_x_BUNDLING` ist 'yes'.

9 – remove channel Abschalten des 2. ISDN-Kanals. Siehe auch "add channel".

Sonst gelten bei diesen Kommandos dieselben Bemerkungen wie für den Windows-imond-Client `imonc.exe`.

Noch zu bemerken ist: Ab fli4l-1.4 ist es nun auch möglich, auf dem fli4l-Router selbst einen "minimalisierten" imon-Client zu installieren, nämlich durch Setzen von `OPT_IMONC`='yes' im Paket `TOOLS`.

Damit kann man nun auch an der fli4l-Konsole bestimmte Einstellungen, z.B. Routing etc. mit imonc vornehmen. Achtung: Dieser Mini-imonc funktioniert nur auf dem fli4l-Router selbst! Auf einem Linux-/Unix- Client ist immer der "große Bruder" unix/imonc zu verwenden.

8. Entwickler-Dokumentation

8.1. Inkompatibilitäten zwischen 2.0 und 3.x

Beim Umstellen von Paketen von Version 2.0 auf 3.x sind folgende Dinge zu beachten:

- 3.x verwendet ein anderes Format für die Datei opt/<package>.txt (in 2 Minuten angepaßt), siehe Liste der zu kopierenden Dateien (Seite 303)
- uClibc statt libc5, ausführbare Programme müssen neu gebaut werden, siehe Compilieren von Programmen (Seite 299)
- Startup-Scripte müssen jetzt rc<nummer>.<name> genannt werden, sonst werden sie nicht ausgeführt, siehe Weitere Dateien (Seite 324)
- iptables statt ipchains, Scripte, die ipchains Kommandos verwenden, müssen neu geschrieben werden
- Linux 2.4.x und 2.6.x statt 2.2.x, Kernelmodule müssen ersetzt werden

8.2. Allgemeine Regeln

Damit ein neues Paket in die OPT-Datenbank auf der FLI4L-Homepage aufgenommen wird, müssen einige Regeln beachtet werden. Pakete die sich nicht an diese Regeln halten, können ohne Vorwarnung aus der Datenbank entfernt werden.

1. KEINE Kopieraktionen von Seiten des Benutzers! FLI4L bietet ein ausgefeiltes System, um die Daten der FLI4L-Pakete in das opt.img einzupacken. Alle Dateien, die auf den Router sollen, liegen in opt/
2. Pakete richtig packen und komprimieren: Die Pakete müssen so aufgebaut sein, daß sie sich mühelos ins FLI4L-Verzeichnis entpacken lassen.
3. Die Pakete sollen sich VOLLSTÄNDIG über die Konfigurationsdatei konfigurieren lassen. Ein weiteres Bearbeiten der Konfigurationsdateien darf nicht vom Benutzer verlangt werden. Schwierige Entscheidungen dem Benutzer abnehmen oder in einen erweiterten Bereich verlagern (Ans Ende der Config-Datei mit einem dicken Hinweis: ONLY MODIFY IF YOU KNOW WHAT YOU DO).
4. Noch ein Hinweis zur Konfigurationsdatei: Anhand des Namens einer Variablen muß sich eindeutig erkennen lassen, zu welchem OPT sie gehört. So gehören z.B. zum `OPT_HTTPD` die Variablen `OPT_HTTPD`, `HTTPD_USER_N`, usw.
5. Bitte, bitte, macht kleine Binaries (Programme)! Wenn ihr sie selbst kompiliert habt, vergesst nicht, die Dateien zu strip-en, die besten Ergebnisse lassen sich mit dem folgenden Befehl erzielen:

```
strip -R .comment -R .note <Dateiname>
```

Manchmal hilft dieser Befehl auch bei vorkompilierten Programmen weiter. Nichts ist frustrierender als ein Download von 2MB, wo man nachher feststellt, daß 500kB gereicht hätten...

Wichtig: *Bitte auf diese Art und Weise keine Module strip-en. Sie werden hinterher nicht mehr funktionieren! Wie man Module am sinnvollsten von Ballast befreit, könnt Ihr in Eurem fli4l-Verzeichnis unter src, kernel-2.4 in der Datei mkkernel-strip.sh nachlesen.*

6. Prüft euer Copyright! Wenn ihr Dateien als Vorlagen benutzt, achtet bitte darauf, das Copyright entsprechend zu ändern. Dies gilt besonders für die Config-, Check- und Opt-Textdateien. Ersetzt hier das Copyright durch euren eigenen Namen. Bei wortwörtlich kopierter Dokumentation muß natürlich das Copyright des Orginal-Autors erhalten bleiben!

7. Bitte als Archivtypen nur verbreitete, freie Formate benutzen. Dazu gehören:
   - ZIP (.zip)
   - GZIP (.tgz oder .tar.gz)

   Folgende Formate bitte nicht verwenden:
   - RAR
   - ACE
   - BlackHole
   - LHA
   - etc.

   Auch Windows-Installer-Dateien (.msi) oder selbstextrahierende Archive und Installer (.exe) sind nicht zu benutzen.

8.3. Compilieren von Programmen

Die für das Compilieren von Programmen erforderliche Umgebung wird in einem separaten Archiv angeboten. Dazu bitte die Beschreibung im src.tar.gz ansehen.

8.4. Verwendung eines modifizierten Kernels — Compilieren eigener Module

Möchte man die Konfiguration des 2.4er Kernel (analog für den 2.6er Kernel) für fli4l modifizieren, kann man dazu das Script mkkernel.sh in src/kernel-2.4 aus dem Src-Archive src.tar.gz verwenden. Dieses automatisiert das Bauen des Kerns.

Wie das geschieht, sieht man bei einem Aufruf der Hilfe zum Script:

```
mkkernel.sh -h
```

Will man die Konfiguration verändern, sind folgende Schritte notwendig:

1. Alle evtl. notwendigen zusätzlichen Patches werden nach src/kernel-2.4 kopiert. Die Namen müssen dabei wie folgt lauten:

   ```
   patch_<patch name>_<kernel version>
   ```

2. Den Kern für das Konfigurieren vorbereiten:

   ```
   cd src/kernel-2.4
   mkkernel.sh -fetch  # hole den kern von www.kernel.org
   mkkernel.sh -unpack # packe ihn aus
   mkkernel.sh -patch  # wende die Patches an
   ```

   Hat man den Kern bereits lokal irgendwo liegen, kann man sich das fetch sparen und kann das Verzeichnis, in dem das Archiv liegt, als Option an mkkernel.sh mitgeben.

3. Kernel konfigurieren:

   ```
   cd linux-<kernel version>
   make xconfig #  oder config oder menuconfig
   cp .config ../dot-config
   ```

   Hat man bereits eine fertige dot-config, kann man sich 2. und 3. ganzen Schritt sparen.

4. Den Kern bauen:

   ```
   mkkernel.sh -remove # lösche alte Binaries
   mkkernel.sh -kernel # bau den Kern und seine Module
   mkkernel.sh -hisax  # bau die Hisax-Module
   ```

   Danach liegen der Kern und die Module in einer fli4l-konformen Verzeichnisstruktur unter ./out. Sollte sich die Kernelversion geändert haben, ist in der Datei config/base.txt die Variable KERNEL_VERSION anzupassen, da mkfli4l diese Variable verwendet, um die korrekte Version der Module einzupacken.

Einige andere sinvolle Optionen zu mkkernel.sh sind die folgenden:

- **-all** – führt alle notwendigen Aktionen aus
- **-nohisax** – wie all, nur ohne isdn module
- **-other** – baut die nicht im kernel enthaltenen Module
- **-header** – stellt die Linux-Header zur Verfügung, die man benötigt, um externe Module zu bauen

8.5. Modulkonzept

fli4l wird seit der Version 2.0 in Module (Pakete) aufgeteilt, z.B.

- fli4l-3.6.2 <— Das Basis-Paket
- dns-dhcp
- dsl
- isdn
- sshd
- und viele weitere...

Mit dem Basis-Paket ist fli4l ein reiner Ethernet-Router. Für ISDN und/oder DSL ist das Paket isdn und/oder dsl in dem fli4l-Verzeichnis auszupacken. Entsprechendes gilt für die anderen Pakete.

8.5.1. mkfli4l

Aus den Paketen wird in Abhängigkeit von der konkreten Konfiguration eine Konfigurationsdatei namens rc.cfg und zwei Archive namens rootfs.img und opt.img erstellt, die alle Konfigurationsinformationen und alle benötigten Dateien enthalten. Diese Dateien werden mit Hilfe von mkfli4l erzeugt, welches die einzelnen Pakete einliest und auf Fehler in der Konfiguration prüft.

mkfli4l akzeptiert die in Tabelle 8.1 angegebenen Parameter. Fehlen sie, werden die in Klammern angegebenen default-Werte genommen. Eine vollständige Liste der Optionen (Tabelle 8.1) erhält man, wenn man

```
mkfli4l -h
```

aufruft.

8.5.2. Aufbau

Ein Paket kann mehrere OPTs enthalten, wenn es aber nur eins enthält, ist es allerdings zweckmäßig, das Paket genauso wie das OPT zu nennen. Im folgenden ist <PACKAGE> durch den jeweiligen Paket-Namen zu ersetzen. Ein Paket besteht aus folgenden Teilen:

- Verwaltungsdateien
- Dokumentation
- Entwickler-Dokumentation
- Client-Programme
- Quellcode
- Weitere Dateien

Die einzelnen Teile sind im folgenden näher beschrieben.

Option	Bedeutung	
-c, --config	Setzen des Verzeichnisses, in dem mkfli4l die config-Dateien der Pakete sucht (default: config)	
-x, --check	Setzen des Verzeichnisses, indem mkfli4l die zum Prüfen der Pakete benötigten Dateien sucht (<package>.txt, <package>.exp und <package>.ext; default: check)	
-l, --log	Setzen der Logdatei; mkfli4l protokolliert Fehlermeldungen und Warnungen in dieser Datei (default: img/mkfli4l.log)	
-p, --package	Angabe der Pakete, die geprüft werden sollen, diese Option kann mehrmals angegeben werden, wenn man mehrere Pakete im Zusammenhang prüfen will. Bei Verwendung von -p wird allerdings grundsätzlich zuerst die Datei <check_dir>/base.exp eingelesen, um die allgemeinen reguläeren Ausdrücke, die vom Basis-Paket bereitgestellt werden, zur Verfügung zu stellen. Diese Datei muß also existieren.	
-i, --info	Gibt Auskunft über den Verlauf der Prüfung (welche Dateien werden gelesen, welche Prüfungen werden durchgeführt, welche besonderen Dinge traten während des Prüfprozesses auf)	
-v, --verbose	Ausführlichere Variante von -i	
-h, --help	Zeigt die Hilfe an	
-d, --debug	Gestattet das debuggen des Prüfprozesses. Dies ist als Hilfe für Paketentwickler gedacht, die etwas genauer wissen möchten, wie die Prüfung des Paketes abläuft.	
	Debugoption	**Bedeutung**
	check	show check process
	zip-list	show generation of zip list
	zip-list-skipped	show skipped files
	zip-list-regexp	show regular expressions for ziplist
	opt-files	check all files in opt/<package>.txt
	ext-trace	show trace of extended checks

Tabelle 8.1.: Parameter für mkfli4l

8.5.3. Die Konfiguration der Pakete

In der Datei config/<PACKAGE>.txt werden vom Benutzer Änderungen an der Konfiguration des Pakets vorgenommen. Alle Variablen eines OPTs sollten einheitlich mit dem Namen des OPTs beginnen, also zum Beispiel:

```
#-----------------------------------------------------------------
# Optional package: TELNETD
#-----------------------------------------------------------------
OPT_TELNETD='no'        # install telnetd: yes or no
TELNETD_PORT='23'       # telnet port, see also FIREWALL_DENY_PORT_x
```

Auch hier sind Kommentare und Leerzeilen erlaubt. Alle Werte hinter dem Gleichheitszeichen müssen in Hochkommas[1] eingefasst werden, da es sonst beim Booten zu Problemen kommen kann.

Variablen, die benutzt werden, werden in die rc.cfg übernommen, alles andere wird ignoriert. Einzige Ausnahme sind Variablen mit dem Namen <PACKAGE>_DO_DEBUG. Diese dienen zum Debuggen von Paketen und werden pauschal übernommen.

8.5.4. Die Liste der zu kopierenden Dateien

Die Datei opt/<PACKAGE>.txt enthält Anweisungen, die beschreiben

- welche Dateien zu welchem OPT gehören,
- wann sie in das zu generierende opt.img bzw. ins rootfs übernommen werden sollen,
- welche uid, gid und Rechte sie bekommen soll,
- welche Konvertierungen vor Aufnahme ins tar file erfolgen sollen.

mkfli4l generiert basierend darauf eine Liste mit allen für das Erstellen des Tarballs notwendigen Informationen.

Leere Zeilen und Zeilen, die mit '#' anfangen, werden ignoriert. In einer der ersten Zeile sollte die Version des Opt-File-Formats wie folgt stehen:

```
<erste Spalte>      <zweite Spalte> <dritte Spalte>
opt_format_version  1                   -
```

Die restlichen Zeilen haben folgende Syntax:

```
<erste Spalte>  <zweite Spalte> <dritte Spalte> <folgende spalten>
Variable        Wert            Datei           Optionen
```

1. In der ersten Spalte steht der Name einer Variable, von deren Wert das Übernehmen der in der dritten Spalte stehenden Datei abhängt. Der Name einer Variable kann beliebig oft in der ersten Spalte auftauchen, falls mehrere Dateien von ihr abhängen. Jede Variable, die im opt/<package>.txt File auftaucht, wird von mkfli4l markiert.

[1] Es können sowohl einfache Hochkommas als auch doppelte Hochkommas verwendet werden. Man kann also *FOO='bar'* oder auch *FOO="bar"* schreiben. Die Verwendung von doppelten Hochkommas sollte allerdings die Ausnahme sein und man sollte sich vorher unbedingt darüber informieren, wie eine Unix-Shell mit einfachen und doppelten Hochkommatas umgeht.

2. In der zweiten Spalte steht ein Wert. Stimmt die in der ersten Spalte stehende Variable mit diesem Wert überein und ist die Variable aktiv (d.h. falls die Variable von einer opt-Variable abhängig ist, hat diese den Wert 'yes'), wird die Datei in der dritten Spalte übernommen. Steht eine %-Variable in der ersten Spalte, wird über alle Indizes iteriert und geprüft, ob die jeweilige Variable mit dem Wert übereinstimmt. Ist das der Fall, wird kopiert. Zusätzlich wird vermerkt, daß aufgrund des aktuellen Wertes der Variable eine Datei kopiert wurde.

3. In der dritten Spalte steht der Name einer Datei. Die Pfadangabe erfolgt relativ zum opt-Verzeichnis. Die Datei muß existieren und lesbar sein, sonst gibt es beim Generieren der Floppy einen Fehler und die Generierung wird abgebrochen.

   Beginnt der Dateiname mit einem `rootfs:-` bzw `rootfs:files/`-Präfix, wird die Datei in die Liste der ins rootfs aufzunehmenden Dateien übernommen. Der Präfix wird vorher entfernt.

   Liegt die Datei unterhalb der aktuellen Config-Directories, wird es in die Liste der aus dem Config-Verzeichnis zu übernehmen Files aufgenommen, andernfalls wird die unter opt liegende Datei genommen. Die Datei darf dann kein `rootfs:`-Präfix haben.

   Ist die zu kopierende Datei ein Kernel-Modul, kann man die konkrete Kernel-Version durch $KERNEL_VERSION ersetzen. mkfli4l nimmt dann die Version aus der Konfiguration und setzt sie hier ein. Dadurch kann man einem Paket Module für verschiedene Kern-Versionen mitgeben und es wird immer die für den Kern richtige Version mit auf den Router kopiert. Für Kernel-Module kann der Pfad auch vollkommen entfallen, mkfli4l findet das Modul anhand von *modules.dep* und *modules.alias* (siehe (Seite 305)).

Option	Bedeutung	Standardwert
uid=	Der Eigentümer der Datei, entweder numerisch oder als Name aus passwd	root
gid=	Die Gruppe der Datei, entweder numerisch oder als Name aus group	root
mode=	Die Zugriffsrechte der Datei	644 (rw-r–r–) für Dateien, 755 (rwxr-xr-x) für Verzeichnisse
flags=	Konvertierungen vor der Aufname ins tar-File	• *utxt* - Konverierung ins Unix-Format • *dtxt* - Konverierung ins Dos-Format • *sh* - Shell-Script, Konverierung ins Unix-Format, entfernen überflüssiger Zeichen
name=	Alternativer Name unter dem das File ins Archiv aufgenommen wird	

Abbildung 8.1.: Optionen für Dateien

4. In den anderen Spalten können die in Abbildung 8.1 aufgeführten Optionen für uid/gid/Rechte der Dateien und Konvertierungen stehen.

   Einige Beispiele:

   - kopiere Datei, wenn `OPT_TELNETD`='yes', setze uid/gid auf root und die Rechte auf 755 (rwxr-xr-x)

     ```
     telnetd     yes     files/usr/sbin/in.telnetd mode=755
     ```

   - kopiere Datei, setze uid/gid auf root, die Rechte auf 555 (r-xr-xr-x) und konvertiere die Datei ins Unix-Format bei gleichzeitigem entfernen aller überflüssigen Zeichen

     ```
     base     yes      etc/rc0.d/rc500.killall mode=555 flags=sh
     ```

   - kopiere Datei, wenn `PCMCIA_PCIC`='i822365', setze uid/gid auf root und die Rechte auf 644 (rw–r–r)

     ```
     pcmcia_pcic i82365 files/lib/modules/${KERNEL_VERSION}/pcmcia/i82365.o
     ```

   - kopiere Datei, wenn eine der *_% Variablen mit dem zweiten Feld übereinstimmt , setze uid/gid auf root und die Rechte auf 644 (rw–r–r)

     ```
     net_drv_%   3c503  3c503.o
     ```

   Die Dateien werden nur kopiert, wenn die oben genannten Bedingungen erfüllt sind und das dazugehörige `OPT_PACKAGE`='yes' gesetzt ist. Welche Opt-Variable dazugehört, wird über die Datei check/<package>.txt beschrieben.

   Wenn im Paket eine Variable referenziert wird, die nicht vom Paket selbst definiert wird, kann es passieren, daß das entsprechende Paket nicht installiert ist. Das würde zu einer Fehlermeldung in mkfli4l führen, da es erwartet, daß alle von opt/<package>.txt referenzierten Variablen definiert sind.

   Um diese Situation korrekt handhaben zu können, wurde die "weak"-Deklaration eingeführt. Sie hat das folgende Format:

   ```
   weak        variable    -
   ```

   Dadurch wird die Variable definiert, wenn sie nicht bereits vorhanden ist und ihr Wert wird auf 'undefined' gesetzt.

Konfigurations-spezifische Dateien

In manchen Situationen möchte man originale Dateien durch konfigurations-spezifische Dateien im opt.img ersetzen, wie z.B. Host-Keys, eigene Firewall-Scripte, ... mkfli4l unterstützt dieses Scenario, indem es prüft, ob eine zu kopierende Datei im Konfig-Verzeichnis zu finden ist und übernimmt in diesem Falle diese Datei in die Liste der ins opt.img aufzunehmenden Dateien.

Eine weitere Möglichkeit, konfigurations-spezifische Dateien ans Opt-Archiv anzuhängen wird im Abschnitt Erweiterte Prüfungen der Konfiguration (Seite 317) beschrieben.

Automatische Auflösung von Abhängigkeiten für Kernel-Module

Kernel-Module bauen unter Umständen auf anderen Kernel-Modulen auf. Diese müssen vor ihnen geladen werden und daher gleichfalls in das Archiv aufgenommen werden. *mkfli4l* bestimmt diese Abhängigkeiten anhand von *modules.dep* und *modules.alias* (zweier beim Kernel-Bauen generierter Dateien) und nimmt automatisch alle benötigten Module in die Archive auf. So führt z.B. folgender Eintrag

```
net_drv_%   ne2k-pci    ne2k-pci.o
```

dazu, daß sowohl 8390.o, als auch crc32.o ins Archiv aufgenommen werden, da ne2k_pci von beiden abhängt.

Die notwendigen Einträge auf *modules.dep* und *modules.alias* werden in das rootfs mit aufgenommen und können von *modprobe* zum laden der Treiber genutzt werden.

8.5.5. Die Prüfung der Konfiguration-Variablen

Mit Hilfe der Datei check/<PACKAGE>.txt können die Inhalte der Variablen auf Gültigkeit überprüft werden. Diese Überprüfung war in früheren Versionen fest im Programm MKFLI4L eingebaut, wurde aber im Zuge der Modularisierung von FLI4L in die Check-Dateien ausgelagert. In dieser Datei ist für jede Variable aus den Config-Dateien eine Zeile vorhanden. Diese Zeilen bestehen aus vier Spalten, welche folgende Funktionen haben:

1. Variable - Diese Spalte gibt den Namen der zu überprüfenden Variable aus der Config-Datei an. Wenn es sich dabei um eine Variable handelt, die mehrmals mit verschiedenen Nummern auftauchen kann, wird an Stelle der Nummer ein Prozentzeichen (%) in den Variablenname eingefügt. Dieses wird immer als '_%_' in der Mitte eines Namens bzw. '_%' am Ende eines Namens verwendet. Der Name kann dabei mehrere Prozentzeichen enthalten, so daß man auch mehrdimensionale Arrays realisieren kann. Dann sollte zwischen den Prozentzeichen allerdings etwas stehen, muß aber nicht, was dann allerdings so seltsamen Namen wie foo_%___% führen würde.

   Oftmals hat man das Problem, daß bestimmte Variablen Optionen beschreiben, die man nur in bestimmten Situationen benötigt. Deshalb können Variablen als optional markiert werden. Optionale Variablen werden mit einem vorangestellten '+' gekennzeichnet. Sie können dann da sein, müssen aber nicht. Arrays können auch mit einem '++' Präfix versehen werden. Steht ein '+' davor, kann das Array da sein oder ganz fehlen. Steht '++' davor, können zusätzlich auch noch einzelne Elemente des Arrays fehlen.

2. `OPT_VARIABLE` - Diese Spalte teilt die Variable einem bestimmten OPT zu. Die Variable wird nur auf Gültigkeit überprüft, wenn die hier angegebene Variable auf 'yes' steht. Gibt es keine Opt-Variable, ist hier ein '-' anzugeben.

3. `VARIABLE_N` - Steht in der ersten Spalte eine Variable mit einem % im Namen wird hier die Variable angegeben, die die Häufigkeit des Auftretens der Variable beschreibt. Ist die Variable mehrdimensional, wird die Häufigkeit des letzten Index beschrieben. Hängt die Variable von einem Opt ab, muß die `VARIABLE_N` vom selben Opt oder von keinem Opt abhängig sein. Ist die Variable von keinem Opt abhängig, darf auch die Variable_n von keinem opt abhängig sein. Gibt es keine Variable_n, ist hier ein '-' anzugeben.

4. VALUE - Diese Spalte gibt an, welche Werte für diese Variable eingegeben werden können. Es sind dabei z.B. folgende Angaben möglich:

Name	Bedeutung
NONE	Es wird keine Überprüfung vorgenommen
YESNO	Die Variable muss “yes” oder “no” sein
NOTEMPTY	Die Variable darf nicht leer sein
NOBLANK	Die Variable darf kein Leerzeichen enthalten
NUMERIC	Die Variable muss numerisch sein
IPADDR	Die Variable muss eine IP-Adresse sein
DIALMODE	Die Variable muss “on”, “off” oder “auto” sein

Werden die Werte mit einem **WARN_**-Präfix versehen, so führt ein illegaler Wert nicht zu einer Fehlermeldung und damit zu einem Abbruch von mkfli4l, sondern nur zur Ausgabe einer Warnung.

Die möglichen Prüfungen werden durch reguläre Ausdrücke in check/base.exp definiert. Diese Datei kann erweitert werden und enhält neuerdings z.B. zusätzlich folgende Prüfungen: HEX, NUMHEX, IP_ROUTE, DISK und PARTITION.

Die Anzahl der Ausdrücke kann jederzeit erweitert werden, hier ist Input von den Opt-Entwicklern erforderlich.

Zusätzlich können reguläre Ausdrücke auch direkt in den check-Dateien angegeben werden, wobei man auch Bezug auf existierende Ausdrücke nehmen kann. Statt YESNO könnte man z.B. auch

```
RE:yes|no
```

schreiben. Sinnvoll ist es dann, wenn ein Test nur ein einziges mal ausgeführt wird und relativ einfach ist. Für genauere Informationen siehe nächstes Kapitel.

Die Sache mit dem Prozentzeichen läßt sich am Besten mit einem Beispiel erklären. Nehmen wir an, in der check/base.txt steht:

```
NET_DRV_N          -                -                NUMERIC
NET_DRV_%          -                NET_DRV_N        NONE
NET_DRV_%_OPTION   -                NET_DRV_N        NONE
```

Das heißt, dass je nach Wert von `NET_DRV_N` die Variablen `NET_DRV_N`, `NET_DRV_1_OPTION`, `NET_DRV_2_OPTION`, `NET_DRV_3_OPTION`, usw. überprüft werden.

8.5.6. Eigene Definitionen zum Prüfen der Konfigurationsvariablen

Einführung regulärer Ausdrücke

In der Version 2.0 gab es nur die oben angeführten 7 Werte-Bereiche, auf die Variablen geprüft werden können: None, NotEmpty, Numeric, IPAddr, YesNo, NoBlank, Dialmode. Die Überprüfung war in mkfli4l fest eingebaut, nicht erweiterbar und beschränkt sich auf wesentliche “Datentypen”, die mit vertretbarem Aufwand geprüft werden können.

Mit der Verion 2.1 wurde diese Prüfung neu implementiert. Ziel der neuen Implementierung ist eine flexiblere Prüfung der Variablen, die auch in der Lage ist, komplexere Ausdrücke zu prüfen. Deshalb werden reguläre Ausdrücke verwendet, die in einem oder

mehreren separaten Dateien abgespeichert werden. Dadurch wird es zum einen möglich, Variablen zu prüfen, die im Augenblick noch nicht geprüft werden und zum anderen können Entwickler optionaler Pakete eigene Ausdrücke definieren, um die Konfiguration ihrer Pakete prüfen zu lassen.

Eine Beschreibung regulärer Ausdrücke findet man via “man 7 regex” oder z.B. hier: http://unixhelp.ed.ac.uk/CGI/man-cgi?regex+7.

Spezifikation regulärer Ausdrücke

Spezifizieren kann man die Ausdrücke auf zwei Wegen:

a) Paketspezifische *.exp Datei check/<PACKAGE>.exp

Diese Datei liegt im check-Direktory und trägt den gleichen Namen wie das dazugehörige Paket, Also z.B. base.exp. Es enthält Definitionen für Ausdrücke, die im *.txt Dateien referenziert werden können. So enthält das base.exp im Augenblick Definitionen für die bekannten Prüfungen und isdn.exp eine Definition für die Variable `ISDN_CIRC_?_ROUTE` (das Fehlen dieser Überprüfung war der Auslöser dieser Änderungen).

Die Syntax lautet wie folgt, wobei man auch hier bei Bedarf doppelte Hochkommatas verwenden kann:

```
Name = 'Regulärer Ausdruck' : 'Fehlermeldung'
```

oder am Beispiel aus base.exp:

```
NOTEMPTY = '.*[^ ]+.*'          : 'should not be empty'
YESNO    = 'yes|no'             : 'only yes or no are allowed'
NUMERIC  = '0|[1-9][0-9]*'      : 'should be numeric (decimal)'
OCTET    = '1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5]'
         : 'should be a value between 0 and 255'
IPADDR   = '((RE:OCTET)\.){3}(RE:OCTET)' : 'no valid ip address'
EIPADDR  = '()|(RE:IPADDR)'
         : 'should be empty or contain a valid ip address'
NOBLANK  = '[^ ]+'              : 'should not contain spaces'
DIALMODE = 'auto|manual|off'    : 'only auto, manual or off are allowd'
NETWORKS = '(RE:NETWORK)([[:space:]]+(RE:NETWORK))*'
         : 'no valid network specification, should be one or more
            network addresse(s) followed by a netmask,
            for instance 192.168.6.0/24'
```

In den regulären Ausdrücken können auch Referenzen auf bereits existierende Definitionen enthalten sein. Diese werden dann einfach an der Stelle eingefügt. Dadurch ist es einfacher, reguläre Ausdrücke zu konstruieren. Eingefügt werden die Referenzen einfach durch '(RE:Referenz)'.

Die Fehlermeldungen tendieren dazu, zu lang zu werden. Daher besteht die Möglichkeit, sie über mehrere Zeilen zu verteilen. Die folgenden Zeilen müssen dann immer mit einem Whitespace (Leerzeichen, Tabulator) beginnen. Beim Einlesen der package.exp-Datei werden überflüssige Whitespaces auf eins reduziert und Tabulatoren durch Whitespaces ersetzt. Ein Eintrag in die *.exp Datei könnte dann so aussehen:

```
NUM_HEX         = '0x[[:xdigit:]]+'
                : 'should be a hexadecimal number
                   (a number starting with "0x")'
```

b) Reguläre Ausdrücke direkt in der Check-Datei in check/*.txt

Manche Ausdrücke kommen nur einmal vor, dann lohnt es sich nicht, dafür einen regulären Ausdruck in *.exp zu definieren. Dann kann man diesen Ausdruck einfach in die Check-Datei schreiben, z.B:

```
# Variable      OPT_VARIABLE    VARIABLE_N      VALUE
MOUNT_BOOT      -               -               RE:ro|rw|no
```

`MOUNT_BOOT` kann lediglich die Werte 'ro', 'rw' oder 'no' annehmen, alles andere wird abgelehnt.

Will man Bezug auf existierende reguläre Ausdrücke nehmen, fügt man einfach eine Referenz via '(RE:...)' ein.

Erweiterung existierender regulärer Ausdrücke

Fügt ein optionales Paket einen zusätzlichen Wert für eine Variable hinzu, die von einem regulären Ausdruck geprüft wird, muß der reguläre Ausdruck erweitert werden. Dies geschieht einfach durch Definition der neuen möglichen Werte durch einen regulären Ausdruck (wie oben beschrieben) und Ergänzung des bestehenden regulären Ausdrucks in einer eigenen check/<package>.exp-Datei. Daß ein bestehender Ausdruck modifiziert werden soll, kennzeichnet ein führendes '+'. Der neue Ausdruck ergänzt den bestehenden Ausdruck, indem der neue Wert als Alternative an den bestehenden Wert angehängt wird. Verwendet ein anderer Ausdruck den ergänzten Ausdruck, gilt auch dort die Ergänzung. Die angegebene Fehlermeldung wird einfach an die vorhandene hinten angehängt.

Am Beispiel der Ethernet-Treiber könnte das wie folgt aussehen:

- Das Basis-Paket stellt eine Menge von Ethernet-Treibern bereit und prüft die Variable `NET_DRV_x` mit dem regulären Ausdruck `NET_DRV`, der wie folgt spezifiert ist:

```
NET_DRV         = '3c503|3c505|3c507|...'
                : 'invalid ethernet driver, please choose one'
                  ' of the drivers in config/base.txt'
```

- Das PCMCIA-Paket stellt jetzt zusätzliche Gerätetreiber bereit, muß also `NET_DRV` ergänzen. Das sieht dann wie folgt aus:

```
PCMCIA_NET_DRV = 'pcnet_cs|xirc2ps_cs|3c574_cs|...' : ''
+NET_DRV       = '(RE:PCMCIA_NET_DRV)' : ''
```

Nun kann man zusätzlich auch noch pcmcia Treiber auswählen.

Regex anhand von YESNO Variablen definieren

Wenn man NET_DRV wie oben um die PCMCIA Treiber erweitert hat, aber das PCMCIA Paket deaktiviert hat, könnte man dennoch einen PCMCIA Treiber in der base.txt auswählen, ohne das eine Fehlermeldung beim Erstellen der Diskette auftritt. Um das

zu verhindern, kann man die Regex auch abhängig von einer YESNO Variablen in der Konfiguration machen. Dazu wird der Name der Variablen die bestimmt ob die Regex benutzt wird mit runden Klammern direkt hinter den Namen der Regex angehängt. Steht die Variable auf 'yes' wird die regex benutzt, ansonsten wird die regex nicht definiert.

```
PCMCIA_NET_DRV(OPT_PCMCIA) = 'pcnet_cs|xirc2ps_cs|3c574_cs|...' : ''
+NET_DRV       = '(RE:PCMCIA_NET_DRV)' : ''
```

Wird jetzt OPT_PCMCIA='no' gesetzt und in der base.txt wird z.B. der Treiber xirc2ps_cs benutzt gibt es beim Erstellen der Diskette einen Fehler.

Regex in Abhängigkeit von anderen Variablen definieren

Alternativ kann man auch beliebige Werte von Variablen als Bedingung verwenden, die Syntax sieht dann wie folgt aus:

```
+NET_DRV(KERNEL_VERSION=~'^2\.4\..*$') = ...
```

Wenn `KERNEL_VERSION` auf den angegebenen Ausdruck matched, also irgend eine 2.4er Kernelversion genutzt wird, dann wird die Liste der erlaubten Netzwerktreiber um die angegebenen Treiber ergänzt.

Fehlermeldungen

Findet die Prüfung einen Fehler, erscheint eine Fehlermeldung der folgenden Art:

```
Error: wrong value of variable HOSTNAME: '' (may not be empty)
Error: wrong value of variable MOUNT_OPT: 'rx' (user supplied regular expression)
```

Beim ersten Fehler wurde der Ausdruck in einem exp file definiert und ein Hinweis auf den Fehler wird mit ausgegeben. Im zweiten Falle wurde der Ausdruck direkt in der *.txt Datei spezifiziert, deshalb gibt es keinen zusätzlichen Hinweis auf die Fehlerursache.

Definition regulärer Ausdrücke

Reguläre Ausdrücke sind wie folgt definiert:

Regulärer Ausdruck: Eine oder mehrere Alternativen, getrennt durch '|', z.B. 'ro|rw|no'. Trifft eine der Alternativen zu, trifft der ganze Ausdruck zu (hier wären 'ro', 'rw' und 'no' gültige Ausdrücke).

Eine Alternative ist eine Verkettung mehrerer Teilstücke, die einfach aneinandergereiht werden.

Ein Teilstück ist ein "Atom", gefolgt von einem einzelnen '*', '+', '?' oder '{min, max}'. Die Bedeutung ist wie folgt:

- 'a*' - beliebig viele a's einschließlich kein a

- 'a+' - mindestens ein a, sonst beliebig viele a's
- 'a?' - kein oder ein a
- 'a{2,5} - zwei bis 5 a's
- 'a{5} - 5 a's
- 'a{2,} - mindestens 2 a's

Ein "Atom" ist ein

- regulärer Ausdruck eingeschlossen in Klammern, z.B. (a|b)+ trifft auf eine beliebige Zeichenkette zu, die mindestens ein a oder b enthält, sonst aber beliebig viele und in beliebiger Reihenfolge
- ein leeres Paar Klammern steht für einen "leeren" Ausdruck
- ein Ausdruck mit eckigen Klammern '[]' (siehe weiter unten)
- ein Punkt '.', der auf irgend ein einzelnes Zeichen zutrifft, z.B. '.+' trifft auf eine beliebige Zeichenkette zu, die mindestens ein Zeichen enthält
- ein '^ ' steht für den Zeilenanfang, z.B. '^ a.*' trifft auf eine Zeichenkette zu, die mit einem a anfängt und in der beliebige Zeichen folgen, 'a' oder 'adkadhashdkash'
- ein '$' steht für das Zeilenende
- ein '\' gefolgt von einem der Sonderzeichen '^ .[$()|*+?{'' steht für genau das zweite Zeichen ohne seine spezielle Bedeutung
- ein normales Zeichen trifft auf genau das Zeichen zu, z.B. 'a' trifft auf genau 'a' zu.

Ein Ausdruck mit rechteckigen Klammern bedeutet folgendes

- '[x-y]' - trifft auf irgend ein Zeichen zu, das zwischen x und y liegt, z.B. '[0-9]' steht für alle Zeichen zwischen 0 und 9; '[a-zA-Z]' für alle Buchstaben, egal ob groß oder klein
- '[^ x-y]' - trifft auf irgendein Zeichen zu, das nicht im angegebenen Intervall liegt
- '[:character_class:] - trifft auf ein Zeichen der Zeichen-Klasse zu. Relevante Standardzeichenklassen sind: alnum, alpha, blank, digit, lower, print, punct, space, upper, xdigit.

Beispiele für Reguläre Ausdrücke

Sehen wir uns das mal an einigen Beispielen an:

Numerisch: Ein numerischer Wert besteht aus mindestens einer, aber beliebig vielen Zahlen. Mindestens ein, aber beliebig viele drückt man mit '+' aus, eine Zahl hatten wir schon als Beispiel. Zusammengesetzt ergibt das:

```
NUMERIC = '[0-9]+' oder alternativ
NUMERIC = '[[:digit:]]+'
```

NOBLANK: Ein Wert, der keine Leerzeichen enthält ist ein beliebiges Zeichen (außer dem Leerzeichen) und davon beliebig viele:

```
NOBLANK = '[^ ]*'
```

bzw. wenn der Wert zusätzlich auch nicht leer sein soll:

```
NOBLANK = '[^ ]+'
```

Disk und Partition: Gültige Bezeichner für eine Disk beginnen mit hd bei IDE-Disks bzw sd bei SCSI-Disks. Dann folgen Buchstaben von a-z (a für die erste Disk, b für die 2., ...) und bei Partitionen die Zahlen 1-8 (1-4 für die ersten 4 Partitionen, die Primär bzw. Extended (nur eine) sein können und 5-8 für die logischen Partitionen innerhalb einer extended Partition). Die Ausdrücke sehen dann wie folgt aus:

```
DISK      = '(hd|sd)[a-z]'
PARTITION = '(hd|sd)[a-z][1-8]'
```

Sehen wir uns das ganze nochmal am Beispiel der IP-Addresse an. Eine IP-Adresse besteht aus 4 Octets, die mit einem '.' getrennt sind. Ein Octet kann eine Zahl zwischen 0 und 255 sein. Definieren wir als erstes ein Octet. Es kann

eine Zahl zwischen 0 und 9 sein:	[0-9]
eine Zahl zwischen 00 und 99:	[0-9][0-9]
eine Zahl zwischen 100 und 199:	1[0-9][0-9]
eine Zahl zwischen 200 und 249:	2[0-4][0-9]
eine Zahl zwischen 250 und 255 sein:	25[0-5]

Da sich die ersten drei Teile stark ähneln, kann man sie zusammenfassen:

- ein Zahl zwischen 0 und 199: 1?[0-9]?[0-9] (die ersten beiden Stellen können, aber müssen nicht da sein)

Das ganze sind Alternativen, also fassen wir sie einfach mittels '|' zu einem Ausdruck zusammen: '1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5]' und haben damit ein Octet. Daraus können wir nun eine IP-Adresse machen, 4 Octets mit Punkten voneinander getrennt (der Punkt muß mittels *backslash* gequotet werden, da er sonst für ein beliebiges Zeichen steht). Basierend auf der Syntax der Exp-Dateien sieht das ganze dann wie folgt aus:

```
OCTET  = '1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5]'
IPADDR = '((RE:OCTET)\.){3}(RE:OCTET)'
```

Unterstützung beim Entwurf regulärer Ausdrücke

Will man reguläre Ausdrücke entwerfen und testen, kann man dazu das regexp tool verwenden, das sich in base/{unix,window}/regexp befindet. Es akzeptiert die folgende Syntax:

```
usage: regexp [-c <check dir>] regexp string
```

Dabei bedeuten die Parameter folgendes:

- <check dir> ist das Verzzeichnis, das die check-Dateien und damit auch die *.exp Dateien enthält. Diese werden von regexp einglesen, damit man auf bereits definiert Ausdrücke zurückgreifen kann.
- regexp ist der reguläre Ausdruck (im Zweifelsfall immer in " angeben)
- string ist der zu prüfende String

Das könnte z.B. wie folgt aussehen:

```
./regexp -c ../check '[0-9]' 0
./regexp -c ../check '[0-9]' a
Error: wrong value of variable command_line_term: 'a'
       (expression didn't match)
./regexp -c ../check IPADDR 192.168.0.1
./regexp -c ../check IPADDR 192.168.0.256
Error: wrong value of variable command_line_term:
       '192.168.0.256' (no valid ip address)
```

8.5.7. Erweiterte Prüfungen der Konfiguration

Manchmal ist es notwendig, komplexere Überprüfungen durchzuführen. Beispiele für solche komplexeren Dinge wären z.B. Abhängigkeiten zwischen Paketen oder Bedingungen, die nur erfüllt sein müssen, wenn Variablen bestimmte Werte annehmen. So muß z.B. bei Auswahl eines PCMCIA-ISDN-Adapters auch das PCMCIA-Paket installiert werden.

Um diese Überprüfungen durchführen zu können, kann man in check/<PACKAGE>.ext kleinere Tests schreiben. Die Sprache besteht aus folgenden Elementen:

1. Schlüsselwörter:
   - Kontrollfluß:
     - if (expr) then statement else statement fi
     - foreach var in set_var do statement done
     - foreach var in set_var_1 ... set_var_n do statement done
     - foreach var in var_n do statement done
   - Abhängigkeiten:
     - provides package version x.y.z
     - depends on package version x.y[<regulärer ausdruck>]
   - Aktionen:
     - warning "warning"
     - error "error"
     - fatal_error "fatal error"
     - set var = value
     - stat (filename, res)
     - crypt (variable)
     - split (string, set_variable, character)

2. Datentypen: strings, numerische Werte, Versionsnummern, Arrays
3. Logische Operationen: <, ==, >, !=, !, &&, ||, =~, copy_pending, samenet, subnet

Definition eines Paket-Identifiers mit einer dazugehörenden Versionsnummer: provides

Damit kann z.B. ein Paket deklarieren, daß es einen Drucker-Dienst bereitstellt oder ein httpd-Paket, daß es einen Webserver bereitstellt. Es kann jeweils nur ein einziges Paket geben, daß einen Identifier bereitstellt. Damit kann man verhindern, daß z.B. zwei Webserver parallel installiert werden, was naheliegenderweise nicht gehen würde, da sich die beiden Server um den Port 80 streiten würden. Zusätzlich wird die aktuelle Version des Pakets angegeben, so daß Weiterentwicklungen Rechnung getragen werden kann. Die Versionsnummer ist dreigeteilt, die ersten beiden Teile sind einstellig, der letzte dreistellig, z.B. 2.1.23.

```
provides IDENTIFIER version VERSION_NUMBER
```

Definition einer Abhängigkeit von einem anderen Paket mit einer bestimmten Version: depends

Benötigt man zur Erbringung der eigenen Funktionalität ein anderes Paket (z.B. einen Webserver), kann man hiermit diese Abhängigkeit von einem Paket mit einer bestimmten Version spezifizieren. Die Version kann zweistellig (z.B. 2.1), dreistellig (z.B. 2.1.11) oder als dreistelliger regulärer Ausdruck angegeben werden (z.B. 2.1.1[12], also sowohl 2.1.11 als auch 2.1.12 sind akzeptabel).

```
depends on IDENTIFIER version VERSION
```

Kommunikation mit dem Nutzer: warning, error, fatal_error

Mit Hilfe dieser drei Funktionen kann man Nutzer warnen, einen Fehler signalisieren oder die Prüfung sofort abbrechen. Das Format sieht wie folgt aus:

- warning “text”
- error “text”
- fatal_error “text”

Im Text kann man auf Variablen Bezug nehmen, indem man einfach den Namen mit einem vorangestellten ’$’ oder ’%’ Zeichen in den Text schreibt (evtl. in {} eingeschlossen, um den Variablenamen vom umgebenden Text abzugrenzen). mkfli4l versucht den darauffolgenden Text (Ziffern, Zahlen, ’_’) als Variablennamen zu interpretieren und setzt bei einem vorangestellten ’$’ den Inhalt der Variablen an dieser Stelle ein, wenn Sie definiert ist, bzw bei einem vorangestellten ’%’ den vollständigen Namen der aktuellen %-Variable. Sonst steht der originale Text da. Will man wirklich ein ’$’ oder ein ’%’ im Text haben, schreibt man ’$$’ bzw. ’%%’.

Zuweisungen

Benötigt man aus irgend einem Grund eine temporäre Variable, kann man diese einfach mit "set var [= value]" anlegen. Die Variable darf kein Config-Variable sein. Läßt man den "= value" Teil weg, wird die Variable einfach auf "yes" gesetzt, so daß man sie hinterher einfach in einer if-Anweisung testen kann. Wird ein Zuweisungsteil angegeben, kann hinter dem Gleichheitszeichen alles stehen, normale Variablen, Indizierte Variablen, Zahlen, Zeichenketten, Versionen.

Arrays

Will man auf einzelne Elemente einer %-Variablen (eines Arrays) zugreifen, kann man das wie gewohnt mit %_var[index] tun, wobei für jedes % Zeichen ein [Index] auftauchen muß.

Verschlüsseln eines Passwortes - crypt

Einige Variablen enthalten Passwörter, die nicht im Klartext in der rc.cfg stehen sollen. Diese Variablen können mittels crypt verschlüsselt werden und werden damit in das Format überführt, daß auch auf dem Router benötigt wird. Verwendet wird das wie folgt:

```
crypt (password)
```

Abfragen von Eigenschaften einer Datei - stat

stat() ermöglicht es, Eigenschaften einer Datei abzufragen. Zur Verfügung gestellt wird im Augenblick lediglich die Größe einer Datei, andere Attribute sind aber leicht hinzuzufügen. Wenn man auf Dateien unterhalb des aktuellen config Verzeichnisses testen will, kann man $config$ benutzen um auf das aktuelle config Verzeichnis zuzugreifen. Abfragen sehen wie folgt aus (wobei die verwendeten Parameter nur Beispiele sind):

```
stat ("unix/Makefile", test)

foreach i in openvpn_%_secret
do
   stat("$$config$$/etc/openvpn/$i.secret", keyfile)
   if (keyfile_res != "OK")
   then
      error "OpenVPN: missing secretfile <config>/etc/openvpn/$i.secret"
   fi
done
```

In dem Beispiel mit der Schleife wird geprüft, ob eine Datei mit einem Namen aus einer Configvariablen im aktuellen config Verzeichnis vorhanden ist. Wenn also OPENVPN_1_SECRET='test' in der Konfigurationsdatei gesetzt wird, prüft die Schleife im ersten Durchlauf ob im aktuellen config Verzeichnis die Datei etc/openvpn/test.secret vorhanden ist.

Danach sind zwei Variablen definiert:

- test_res: Resultat des Systemrufs stat() ("OK", wenn Systemruf erfolgreich, sonst Fehlermeldung des Systemrufs)
- test_size: Größe der Datei

Das könnte dann z-B. so aussehen:

```
stat ("unix/Makefile", test)
if ("$test_res" == "OK")
then
        warning "test_size = $test_size"
else
        error "Error '$test_res' while trying to get size of ..."
fi
```

Grep in einer Datei - fgrep

Wenn Sie in einer Datei per grep suchen wollen steht Ihnen das fgrep() Kommando zur Verfügung. Die Syntax ist fgrep(<dateiname>, <regex>). Wenn die Datei <dateiname> nicht existiert wird mkfli4l mit einem fatalen Fehler beendet! Wenn Sie also nicht sicher sind, ob die Datei immer vorhanden ist, testen Sie die Existenz von <dateiname> vorher mit stat() ab. Nach dem Aufruf von fgrep() steht Ihnen das Suchresultat in wie folgt zur Verfügung:

- FGREP_MATCH_N: Hier steht die Anzahl der Treffer die im dem FGREP_MATCH Array stehen.
- FGREP_MATCH_x: Wobei x von 0 bis FGREP_MATCH_N reicht. Dort stehen die einzelnen Treffer des grep.

Das folgende Beispiel zeigt die Verwendung von fgrep() aus einem Beispiel aus der base.ext. Hier wird getestet ob alle in der PF_FORWARD_x angegebenen tmpl: Referenzen vorhanden sind.

```
foreach n in pf_forward_n
do
  set rule=pf_forward_%[n]
  if (rule =~ "tmpl:([^[:space:]]+)")
  then
    foreach m in match_%
    do
      stat("$config_dir/etc/fwrules.tmpl/$m", tmplfile)
      if(tmplfile_res == "OK")
      then
        add_to_opt "etc/fwrules.tmpl/$m"
      else
        stat("opt/etc/fwrules.tmpl/$m", tmplfile)
        if(tmplfile_res == "OK")
        then
          add_to_opt "etc/fwrules.tmpl/$m"
        else
          fgrep("opt/etc/fwrules.tmpl/templates", "^$m[[:space:]]+")
          if (fgrep_match_n == 0)
          then
            error "Can't find tmpl:$m for PF_FORWARD_${n}='$rule'!"
          fi
        fi
      fi
```

```
    done
  fi
done
```

Auseinandernehmen von Parametern - split

Oftmals werden Variablen mir mehreren Parametern belegt, die dann in Startup-Scripten erst wieder auseinandergenommen werden. Will man diese bereits vorher auseinandernehmen und Tests mit ihnen durchführen, nimmt man split.

```
split (string, %_var, character)
```

Der String kann durch eine Variable, %-Variable oder direkt als String angegeben werden. mkfli4l zerlegt ihn an den Stellen, an denen das Trennzeichen auftaucht und erzeugt je eine Instanz der %-Variablen. Über diese kann man dann hinterher Tests laufen lassen. Steht zwischen zwei Trennzeichen nichts, wird eine Instanz mit einer leeren Zeichenkette als Wert erzeugt. Ausnahme ist ' ', hier werden alle whitespaces konsumiert und keine leeren Variablen erzeugt.

Sollen die bei der Zerlegung entstandenen Elemente in einem numerischen Kontext verwendet werden (z.B. als Index), muß das beim Aufruf von Split spezifiert werden. Das geschieht durch das zusätzliche Attribut 'numeric'. Der Aufruf sieht dann wie folgt aus:

```
split (string, %_var, character, numeric)
```

Ein Beispiel könnte wie folgt aussehen:

```
set bar=''1.2.3.4''
split (bar, tmp_%, '.', numeric)
foreach i in tmp_%
do
        warning "%i = $i"
done
```

Hinzufügen von Dateien zum Opt-Archiv

Mit der Funktion `add_to_opt()` können zusätzliche Dateien ans Opt-Archiv angehängt werden. Es können dabei *alle* Dateien unterhalb von opt/ oder aus dem config/ Verzeichnis ausgewählt werden. Eine Beschränkung nur auf die Dateien die mit diesem opt Paket geliefert werden gibt es nicht. Liegt eine Datei sowohl unter opt/ als auch im config/ Verzeichnis im gleichen Pfad vor bevorzugt add_to_opt die Dateien aus dem config/ Verzeichnis. Die Funktion add_to_opt wird in der Regel dann eingesetzt wenn komplexe logische Regeln darüber entscheiden ob und welche Dateien in das opt Archiv kopiert werden müssen.

Die Syntax sieht wie folgt aus: `add_to_opt <filename> [<flags>]`. Im folgenden ein Beispiel aus dem Paket SSHD.

```
if (opt_sshd)
then
   foreach pkf in sshd_public_keyfile_%
   do
     stat("$config_dir/etc/ssh/$pkf", publickeyfile)
     if(publickeyfile_res == "OK")
```

```
    then
        add_to_opt "etc/ssh/$pkf" "mode=400 flags=utxt"
    else
        error "sshd: missing public keyfile ..."
    fi
  done
fi
```

Mit `stat()` (Seite 315) wird zunächst geprüft, ob die Datei im Konfig-Verzeichnis existiert. Ist die Datei vorhanden, wird sie ans Opt-Archiv angehängt, andernfalls bricht `mkfli4l` mit einer entsprechenden Fehlermeldung ab.

Hinweis: Auch bei `add_to_opt` prüft (Seite 305) `mkfli4l`, ob die zu kopierende Datei im Konfig-Verzeichnis zu finden ist.

Kontrollfluss

```
if (expr)
then
        statement
else
        statement
fi
```

Eine klassische if-Konstruktion, wie man sie kennt. Ist die Bedingung wahr, wird der then-Teil ausgeführt, ist die Bedingung falsch, wird der else-Teil ausgeführt.

Will man Tests über %-Variablen durchführen, muß man jede einzelne Variable testen. Dazu gibt es das foreach-Statement in zwei Varianten:

```
foreach loop_var in set_var
do
        statement
done

foreach loop_var in set_var_1 set_var_2 ... set_var_n
do
        statement
done
```

Diese Schleife iteriert über alle %-Variablen, angefangen bei eins bis zum in der dazugehörigen (in check/<PACKAGE>.txt stehenden) Variable_n stehenden Index n. Die Laufvariable loop_var nimmt dabei die jeweiligen Werte der %-Variablen an. Zu beachten ist dabei, dass bei optionalen Variablen, die in der Konfiguration nicht vorhanden sind, eine leere Instanz generiert wird. Unter Umständen muß das im Script berücksichtigt werden, was man z.B. wie folgt tun kann:

```
foreach i in template_var_opt_%
do
    if (i != "")
    then
        warning "%i is present (%i='$i')"
    else
        warning "%i is undefined (empty)"
    fi
done
```

Das Statement in der Schleife kann eine der oben beschriebenen Funktionen if, foreach, provides, depends, warning, error oder fatal_error sein.

Will man genau eine %-Variable testen, kann man diese mittels var_%[index] auswählen. Der index kann dabei eine normale Variable, ein String, oder wiederum ein indiziertes Array sein.

```
foreach loop_var in var_n
do
        statement
done
```

Diese Schleife läuft von 1 bis var_n. Man kann die Schleifenvariable loop_var dazu benutzen, um _% Variablen zu indizieren. Will man also nicht nur über eine _% Variable iterieren, sondern über mehrere gleichzeitig, nimmt man diese Variante der Schleife und verwendet die loop_var zum Indizieren mehrerer _% Variablen.

Expressions

Die Expressions erlaubt so gut wie alles, was man von einer Programmiersprache gewöhnt ist. Ein in einem Ausdruck auftauchender Wert 'val' kann eine Variable, ein String oder ein indiziertes Array sein. Variablen in Strings werden dabei wie oben beschrieben ersetzt. Ein Test auf die Gleichheit zweier Variablen könnte also so aussehen:

```
var1 == var2
"$var1" == "$var"
```

Zu beachten ist dabei, daß der Vergleich in Abhängigkeit vom Typ der Variable erfolgt, der in check/<package>.txt festgelegt wurde. Variablen, die als Typnamen einen mit NUM beginnenden Namen haben erhalten einen numerischen Typ. Ist einer der beiden Variablen numerisch, erfolgt der Vergleich auf numerischer Basis, d.h. die Zeichenketten werden in Zahlen umgewandelt und dann vergleichen. Sonst erfolgt der Vergleich auf String-Basis; ein Vergleich von '05' und '5' geht ergibt ungleich, ein Vergleich von '18' und '9' ergibt '18' < '9'.

Für den Vergleich von Versionen wird das Hilfkonstrukt numeric(version) eingeführt, welches den numerischen Wert für einen Versionsstring für Vergleichszwecke bestimmt. numeric(version) == sub_version + 1000 * minor_version + 10000 * major_version

Eine Vollständige Auflistung aller Ausdrücke ist in Tabelle 8.2 zu finden.

Match-Operator

Mit dem Match-Operator `=~` kann man prüfen, ob ein regulärer Ausdruckt auf den Wert einer Variable "matches". Weiterhin kann man den Operator auch nutzen, um Teilausdrücke aus einer Variablen zu extrahieren. Nach erfolgreichem Anwenden eines regulären Ausdrucks auf eine Variable enthält das Array match_% die gefundenen Teile. Das könnte z.B. wie folgt aussehen:

```
set foo="foobar12"
if ( foo =~ "(foo)(bar)([0-9]*)" )
then
        foreach i in match_%
        do
```

expr	true if
id	id == 'yes'
val == val	strings/numerische Werte sind identisch
val != val	strings/numerische Werte sind verschieden
val == number	numerischer Wert von val == number
val != number	numerischer Wert von val != number
val < number	numerischer Wert von val < number
val > number	numerischer Wert von val > number
val == version	numeric(val) == numeric(version)
val < version	numeric(val) < numeric(version)
val > version	numeric(val) > numeric(version)
val =~ string	regulärer Ausdruck in string auf val paßt
(expr)	Ausdruck in Klammern ist wahr
expr && expr	beide Ausdrücke sind wahr
expr || expr	mind. einer der beiden Audrücke ist wahr
copy_pending(id)	siehe Beschreibung
samenet (string1, string2)	string1 das gleiche netz wie string2 beschreibt
subnet (string1, string2)	string1 ein Subnet von string2 beschreibt

Tabelle 8.2.: Logische Ausdrücke

```
                warning "match %i: $i"
        done
fi
```

Ein mkfli4l aufruf führt dann zu folgender Ausgabe:

```
Warning: match MATCH_1: foo
Warning: match MATCH_2: bar
Warning: match MATCH_3: 12
```

Bei Verwendung von =~ kann Bezug auf alle existierenden regulären Ausdrücke genommen werden. Will man z.B. prüfen. ob ein PCMCIA-Ethernet-Treiber ausgewählt wurde, ohne daß opt_pcmcia auf yes gesetzt wurde, könnte das wie folgt aussehen:

```
if (!opt_pcmcia)
then
    foreach i in net_drv_%
    do
       if (i =~ "(RE:PCMCIA_NET_DRV)")
       then
           error "If you want to use ..."
       fi
    done
fi
```

Prüfen, ob in Abhängigkeit vom Wert einer Variable eine Datei kopiert wurde: copy_pending

Mit den im Check-Prozeß gewonnenen Informationen prüft die Funktion copy_pending() ob in Abhängigkeit vom Wert einer Variable eine Datei kopiert wurde, oder nicht. Das kann man verwenden, um z.B. zu testen, ob der vom Nutzer angegebene Treiber auch wirklich existiert und kopiert wurde. copy_pending akzeptiert den zu prüfenden Namen in Form einer Variablen oder eines Strings. Im String werden Ersetzungen wie oben beschrieben durchgeführt, so daß man z.B. mittels einer foreach Schleife und einem "%loop_var" alle Inkarantionen einer %-Variable prüfen kann.

copy_pending() prüft dazu, ob

- die Variable aktiv ist (wenn sie von einem opt abhängt, muß dieses auf 'yes' gesetzt sein)
- die Variable in einer opt/<package>.txt Datei referenziert wurde
- ob in Abhängigkeit vom aktuellen Wert eine Datei kopiert wurde.

Ein kleines Beispiel für die Anwendung all dieser Funktionen findet man im template package.

Vergleich von Netzwerkadressen – *samenet* und *subnet*

Zum Prüfen von Routen benötigt man ab und zu einen Test, ob zwei Netzwerke identisch sind oder eines ein Subnetz eines anderen ist. Dazu gibt es die beiden Funktionen *samenet* und *subnet*; *samenet (netz1, netz2)* liefert true, wenn beide Netze identisch sind, *subnet (netz1, netz2)* liefert true, wenn netz1 ein Subnetz von netz2 ist.

Erweitern der Kernel-Commandline im ext Script

Ist ein Opt gezwungen, dem Kernel anderen Boot-Parameter zu uebergeben, so musste die Variable KERNEL_BOOT_OPTION geprueft werden, ob der noetige Parameter enthalten war und ggf. ein warning oder error ausgeben. mit der internen Variable KERNEL_BOOT_OPTION_EXT geht das jetzt direkt im ext-script:

```
set kernel_boot_option_ext="${kernel_boot_option_ext} nokbd"
```

aktuell wird dieser Mechanismus von opt_embedded benutzt, um bei den Geraeten WRAP und ALIX den reboot-Mechanismus und die Abwesenheit eines Keyboard-Controllers bekannt zu geben.

8.5.8. Unterstützung von Linux 2.4 und 2.6

Die beiden Kernelversionen unterscheiden sich in kleinen, aber wichtigen Details:

- es stehen andere Treiber zur Verfügung, einige sind weggefallen, andere Hinzugekommen
- die Modul-Namen enden mit '.o' bzw. '.ko'
- die Module heißen teilweise einfach anders
- die Modul-Abhängigkeiten sehen anders aus

- die Module liegen woanders

Diese Unterschiede werden zum großen Teil durch *mkfli4l* automatisch behandelt. Um die zur Verfügung stehenden Module zu beschreiben, kann man zum einen die zur Prüfung verwendeten checks in Abhängigkit von der Version erweitern (bedingte reguläre Ausdrücke (Seite 310)) und zum anderen erlaubt *mkfli4l* versionsabhängige *opt/<package>_2_4.txt* und *opt/<package>_2_6.txt* Dateien, die je nach Version verschiedene Dateien ins Archiv aufnehmen.

8.5.9. Dokumentation

Die Dokumentation wird in den Dateien

- doc/deutsch/opt/<PACKAGE>.txt
- doc/deutsch/opt/<PACKAGE>.html
- doc/english/opt/<PACKAGE>.txt
- doc/english/opt/<PACKAGE>.html

abgelegt. Die HTML-Dateien können auch aufgeteilt werden, d.h. für jedes enthaltene OPT eine. Dann muss trotzdem eine <PACKAGE>.html angelegt werden, die auf die anderen Dateien verweist (siehe Paket HD). Änderungen sollten folgenden Dateien dokumentiert werden:

- doc/deutsch/opt/changes/<PACKAGE>-<VERSION>.txt
- doc/english/opt/changes/<PACKAGE>-<VERSION>.txt

Die gesamte Text-Dokumentation darf keine Tabs enthalten und muß nach spätestens 79 Zeichen einen harten Zeilenumbruch haben. Die stellt sicher, daß die Dokumentation auch mit einem Editor ohne automatischen Zeilen- umbruch richtig gelesen werden kann.

Wer mag kann auch eine Dokumentation im Tex-Format erstellen und daraus dann Text, html, Postscript und PDF-Versionen erzeugen. Als Beispiel kann die Dokumentation von fli4l dienen. Einen Rahmen für die Dokumentation und die minimal benötigten LaTeX-Macros kann man im Template-Paket finden. Eine kurze Beschreibung ist in den folgenden Unterabschnitten zu finden.

Voraussetzungen für die Erstellung einer LaTeX-Dokumentation

Zum Erstellen der Doku aus den LaTeX-Sourcen gibt es folgende Anforderungen an die Umgebung:

- Linux/Unix-Umgebung: Zur einfachen Erzeugung gibt es ein Makefile, mit dem alle weiteren Aufrufe automatisiert sind (Cygwin müsste auch tun, ist aber imho noch nicht getestet)
- installiertes latex2html für die HTML-Version
- Installiertes LaTeX(tetex unter Unix getest) mit pdftex und folgenden Paketen:

- aktuelles KOMA-Skript (Version 2)
- alle notwendigen Pakete für pdftex
- Ausgepacktes Dokumentationspaket für fli4l, welches die benötigten Makefiles und Stylefiles bereitstellt.

Dateinamen

Die Dateien der Dokumentation werden nach folgendem Schema benannt:

<**packagename**>**_main.tex** Enthält den Hauptteil der Dokumentation. <packagename> steht hier für den Namen des Pakets, das beschrieben werden soll (in Kleinbuchstaben). Diese Namen entsprechen den Namen der einzelnen Verzeichnisse unterhalb fli4l.

<**packagename**>**_appendix.tex** Sollen zu diesem Paket noch weitere Anmerkungen im Anhang hinzugefügt werden, so werden diese hier abgelegt.

Diese Dateien werden im Verzeichniss

fli4l/<packagename>/doc/deutsch/tex/<packagename>

abgelegt. Für das sshd Paket sieht das z.B. wie folgt aus:

```
> ls fli4l/doc/deutsch/tex/sshd/
Makefile sshd_appendix.tex  sshd_main.tex  sshd.tex
```

Das Makefile ist für die Generierung der Dokumentation verantwortlich, das sshd.tex File stellt einen Rahmen für die eigentliche Doku und den Anhang bereit, der sich in den anderen beiden Dateien befindet. Ansehen kann man sich das am Beispiel der Template-Dokumentation.

LaTeX-Grundlagen

Disclaimer (Jörg): Ich arbeite selbst noch nicht so lange mit LaTeX, und was ich hier von mir gebe, ist an vielen Stellen sehr vereinfacht und eingeschränkt; wir werden hier längst nicht alle Möglichkeite einsetzen, die LaTeX bietet.

LaTeX arbeitet ähnlich wie HTML "tag-orientiert", nur das die Tags hier Kommandos heissen und folgendes Format aufweisen: `\kommando` bzw. `\begin{umgebung}` ... `\end{umgebung}`

Nach Möglichkeit sollte man die Kommandos lieber logisch einsetzen als als phyische Auszeichnung missbrauchen, also z.B. `\warning{Bitte␣nicht␣...␣tun}` statt `\emph{Bitte␣nicht␣...␣tun}` verwenden.

Jedes Kommando bzw jede Umgebung kann noch weitere Parameter aufnehmen, die mit `\kommando{parameter1}{paramter2}{paramterN}` geschrieben werden.

Optionale Paramter werden wie folgt geschrieben: `\kommando[optionalerParameter]{parameter1}` ..., wobei im Normalfall nur 1 optionaler Parameter vorkommt, in bestimmten Fällen aber auch mehrere.

Einzelne Absätze werden im Dokument-Quelltext durch Leerzeilen getrennt. Innerhalb dieser Absätze nimmt LaTeX selbst den Zeilenumbruch und die Worttrennung vor.

Folgende Buchstaben haben eine spezielle Bedeutung in LaTeX und müssen, sollten sie in normalem Text vorkommen, mit einem vorangestellten \ maskiert werden: # $ & _ % { }. '~' und '^ müssen wie folgt geschrieben werden: `\verb?~?` `\verb?^?`

Die wichtigsten LaTeX-Kommandos werden in der Template-Dokumentation verwendet und erklärt.

8.5.10. Dateiformate

Alle Textdateien (sowohl Dokumentation als auch Scripte, die später auf dem Router liegen) müssen im DOS-Dateiformat, also mit CR-LF statt nur LF in das Paket gelegt werden. Dadurch wird erreicht, daß Windows-User die Dokumentation auch mit dem Notepad lesen können und durch eine Änderung eines Scriptes unter Windows das ganze später auf dem Router trotzdem lauffähig bleibt. Die Scripte werden beim Bauen der Archive in das auf dem Router benötigte Format konvertiert (siehe flags in opt/<package>.txt).

8.5.11. Entwickler-Dokumentation

Sollte ein Programm aus dem Paket eine neue Schnittstelle definieren, die andere Programme nutzen können, so ist die Dokumentation dieser Schnittstelle in einer separaten Dokumentation unter *doc/dev/<PACKAGE>.txt* abzulegen.

8.5.12. Client-Programme

Sollte ein Paket zusätzliche Client-Programme mitliefern, so sind diese im Verzeichnis windows/ für Windows-Clients und im Verzeichnis unix/ für Unix- und Linux-Clients abzulegen.

8.5.13. Quellcode

Angepasste Programme und Quellcodes können im Verzeichnis src/ beigelegt werden. Handelt es sich um mehrere Dateien, sollte zweckmäßigerweise ein entsprechendes Unterverzeichnis angelegt werden. Sollen die Programme genauso wie die restlichen fli4l-Programme gebaut werden, bitte einen Blick in die Dokumentation des Pakets SRC (Seite 260) werfen.

8.5.14. Weitere Dateien

Alle Dateien, die nachher auf dem Router liegen, werden unter opt/etc/ und opt/files/ abgelegt. Dabei liegen unter:

- opt/etc/boot.d und opt/etc/rc.d Srcipte, die beim Booten des Systems ausgefuehrt werden sollen
- opt/etc/rc0.d Scripte, die beim Runterfahren des Systems ausgefuehrt werden
- opt/etc/ppp Scripte, die beim Einwählen und auflegen ausgeführt werden
- opt/files die ausführbaren Programme und sonstige Dateien entsprechend ihrer Positionen im *nix-Filesystem

Die Scripte in opt/etc/boot.d, opt/etc/rc.d und opt/etc/rc0.d werden wie folgt benannt:

rc<nummer>.<name>

Die Nummer entscheidet über die Reihenfolge der Ausführung, die Namen geben einen Hinweis darauf, welches Programm/Paket von diesem Script behandelt wird.

8.6. Allgemeine Skript-Erstellung auf FLI4L

Hier folgt jetzt keine allgemeine Einführung in Shell-Scripts, das kann jeder im Internet selber nachlesen, es wird nur auf die spezielle Gegebenheiten bei FLI4L eingegangen. Infos dazu gibt es in den Unix-/Linux- Manualpages. Folgende Links können als Einstiegspunkte zu diesem Thema dienen:

- Einführung in Shell-Scripts:
  - http://cip.physik.uni-freiburg.de/main/howtos/sh.php
- Manualpages online:
  - http://www.freebsd.org/cgi/man.cgi?manpath=Red+Hat+Linux/i386+9
  - http://heapsort.de/man2web
  - http://man.he.net/
  - http://www.linuxcommand.org/superman_pages.php

Für Unix-User: Leider war es wegen des begrenzten Platzes nicht möglich, erweierte Tools, wie z.B. awk, tail, uniq, etc. einzubinden. Viele von diesen Tools lassen sich aber mit einfachen Maßnahmen nachbilden

8.6.1. Aufbau

In der Unix-Welt ist es nötig, ein Script mit dem Namen des Interpreters zu beginnen, daher steht in der ersten Zeile:

```
#!/bin/sh
```

Damit man später leichter erkennen kann, was ein Script macht und wer es geschrieben hat, sollte jetzt ein kurzer Header folgen, in etwa so:

```
#-----------------------------------------------------------------
# /etc/rc.d/rc500.dummy - start my cool dummy server
#
# Creation:     19.07.2001  Tobias Gruetzmacher <fli4l@portfolio16.de>
# Last Update:  11.11.2001  Tobias Gruetzmacher <fli4l@portfolio16.de>
#-----------------------------------------------------------------
```

Nun kann das eigentliche Script folgen...

8.6.2. Umgang mit Konfigurationsvariablen

Allgemeines

Pakete werden über die Datei <package>/config/<package>.txt konfiguriert. Die darin enthaltenen Variablen werden beim Erzeugen der Boot-Diskette in die Datei rc.cfg übernommen. Beim Booten des Routers wird diese Datei eingelesen, bevor irgend ein rc Script (scripte unter /etc/rc.d) gestartet wird. Diese Scripte können dadurch auf alle Konfigurationsvariablen einfach durch $CONFIG_VARIABLE zugreifen.

Benötigt man Werte von Konfigurationsvariablen auch nach dem Booten noch, müssen sie separat abgelegt werden, da die rc.cfg Datei nach dem Hochfahren des Routers verschwindet. Üblich ist dabei, die Werte in eine Datei unter /var/run/<name> abzulegen. Von dort kann sie dann ein anderes Script wieder auslesen. Dabei gibt es zwei verschiedene Vorgehensweisen:

- Man schreibt die Werte einfach ein eine Datei und ließt sie nachher unter Beachtung des verwendeten Formats aus.
- Man generiert eine eigene Config-Datei, in der die Variablen in der Form FOO='foo' stehen.

Für beides gibt es im folgenden ein konkretes Beispiel.

Beispiel für das Ablegen von Werten in einer Datei

Am Beispiel des ISDN-Paketes erklärt, würde das Auswerten der Konfigurationsvariablen so aussehen (/etc/rc.d/rc340.circuits):

```
# Laufvariable für % Variablen
idx=1
# Ist isdn überhaupt aktiv?
if [ "$OPT_ISDN" = "yes" ]
then
# konfiguriere 1-n circuits
    while [ "$idx" -le "$ISDN_CIRCUITS_N" ]
    do
        # wie ist denn die Einwahlnummer des n-ten circuits
        eval eaz='$ISDN_CIRC_'$idx'_EAZ'
        # und die lokale IP Adresse für das Interface
        eval local='$ISDN_CIRC_'$idx'_LOCAL'
        # ... remote
        eval remote='$ISDN_CIRC_'$idx'_REMOTE'
        # ... netzmaske
        eval netmask='$ISDN_CIRC_'$idx'_NETMASK'
```

Da einige Werte später noch gebraucht werden, werden sie in einer Datei abgelegt:

```
        # Datei erzeugen und Wert reinschreiben
        echo $local >/var/run/$dev.conf
        # Rest anhängen
        echo $remote >>/var/run/$dev.conf
        echo $netmask >>/var/run/$dev.conf
```

Braucht man die Werte später wieder, liest man sie einfach wieder aus, wie man am Beispiel von /usr/local/bin/delete-all-routes sehen kann:

```
# auslesen der abgelegten werte
conf=‘cat /var/run/$dev.conf‘
# die in $conf stehenden Werte werden zu $1, $2, $3, ...
set $conf
# Verwenden der Werte
ip addr add $1 peer $2 dev $device
```

Steht nicht von vornherein fest, wieviele Werte dort drin stehen werden, arbeitet man mit shift, bis keine Variablen mehr da sind. Das könnte wie folgt aussehen:

```
# auslesen der abgelegten werte
conf=‘cat /var/run/$interface.conf‘
# die in $conf stehenden Werte werden zu $1, $2, $3, ...
set $conf
# Verwenden der Werte
while [ "$1" != "" ]
do
    ip route add -net $1 netmask $2 dev $interface
    # die beiden verbrauchten Werte wegwerfen und die nächsten
    # nach vorne holen. Dabei wird in diesem Fall $3 zu $1, $4
    # zu $2, ... , $n zu $(n-2)
    shift 2
done
```

Beispiel für das Ablegen von Variablen im Config-Dateiformat

Möchte man die Variablen im Format der Config-Dateien ablegen, könnte das wir folgt aussehen (am Beispiel von register):

- in /etc/rc.d/rc910.register werden die Config-Variablen in eine Datei geschrieben:

```
{
     echo "register_server='$REGISTER_SERVER'"
     echo "register_always='$REGISTER_ALWAYS'"
} > var/run/register.conf
```

- die Werte werden dann in /usr/local/bin/register.sh einfach über das “source” Komando der Shell (kurzform “. filename”) wieder eingelesen und sind dann einfach in der Umgebung des Scripts verfügbar:

```
. /var/run/register.conf
```

Als Namenskonvention wird hier /var/run/<opt name>.conf festgelegt.

8.6.3. Fehlersuche

Bei Start-Scripts ist es oft sinnvoll, diese bei Bedarf im Debug-Modus der Shell laufen zu lassen, um festzustellen, wo “der Wurm drin ist”. Dazu wird am Anfang und am Ende folgendes eingefügt:

```
begin_script <OPT-Name> "start message"
<script code>
end_script
```

Im normalen Betrieb erscheint jetzt beim Start des Scriptes der angegebene Text und am Ende der gleiche Text mit einem vorangestellten “finished”.

Will man die Scripte debuggen, muß man zwei Dinge tun:

1. Man muß DEBUG_STARTUP (Seite 31) auf 'yes' setzen.
2. Man muß das Debuggen für dieses Opt aktivieren. das tut man in der Regel durch den Eintrag

```
<OPT-Name>_DO_DEBUG='yes'
```

in der Konfigurationsdatei[2]. Jetzt wird während der Ausführung am Bildschirm genau dargestellt, was passiert.

Weitere beim Debuggen hilfreiche Variablen

DEBUG_ENABLE_CORE Diese Variable gestattet das Erzeugen von core-Dumps. Stürzt ein Programm aufgrund eines Fehlers ab, wird ein Abbild des aktuellen Zustandes im Filesystem abgelegt, der hinterher zur Analyse des Problems verwendet werden kann. Die core-Dumps werden unter /var/log/dumps/ abgelegt.

DEBUG_IP Wird diese Variable gesetzt, werden alle Aufrufe des Programms *ip* protokolliert.

DEBUG_IPUP Wird diese Variable auf *yes* gesetzt, werden während der Ausführung der ip-up/-down-Scripte traces mitgezeichnet und im System-Log protokollier.

LOG_BOOT_SEQ Wird diese Variable auf *yes* gesetzt, protokolliert der *bootlogd* während des Bootens alle auf die COnsole getätigten Ausgaben. Diese Variable hat standardmäßig den Wert *yes*.

DEBUG_KEEP_BOOTLOGD Normalerweise wird der bootlogd am Ende des Bootvorganges beendet. Ein Setzten dieser Variable unterbindet das und erlaubt ein Protokollieren der Konsolenausgaben über den Bootvorgang hinaus.

DEBUG_MDEV Ein Setzen dieser generiert ein Protokoll des mdev-Daemons, der für das Anlegen der Geräte-Dateien unter /dev zuständig ist.

8.6.4. Hinweise

- Es ist **immer** besser, geschweifte Klammern an Stelle von runden Klammern () zu benutzen. Allerdings muß dabei darauf geachtet werden, daß nach der öffnenden Klammer ein Leerzeichen oder eine neue Zeile vor dem nächsten Befehl kommt und vor der schließenden Klammer ein Semikolon oder auch eine neue Zeile kommt. Beispielsweise ist:

  ```
  { echo "cpu"; echo "quit"; } | ...
  ```

  gleichbedeutend mit:

  ```
  {
          echo "cpu"
          echo "quit"
  } | ...
  ```

- Ein Script kann mit “exit” vorzeitig beendet werden. Dies ist aber bei den Start-Scripts (opt/etc/boot.d, opt/etc/rc.d/*), den Stop-Scripts (opt/etc/rc0.d/*) und den ip-up/ip-down-Scripts (opt/etc/ppp/*) geradezu tödlich, da auch nachfolgende Scripts nicht mehr ausgeführt werden. Im Zweifelsfall immer weglassen.

[2] Manchmal werden mehrere Startupscripte verwendet, die dann auch verschiedene Namen für ihre Debug-Variablen haben. Hier hilft ein kurzer Blick in die Scripte

- KISS - Keep it small and simple - Du willst Perl als Script-Sprache verwenden? Dir reichen die Scripting-Möglichkeiten von FLI4L nicht? Überdenke deine Einstellung! Ist dein OPT wirklich nötig? FLI4L ist immer noch "nur" ein Router, ein Router sollte eigentlich keine Serverdienste anbieten.
- Die Fehlermeldung ": not found" heißt meistens, das das Script noch im DOS-Format vorliegt. Weitere Fehlerquelle: Das Script ist nicht ausführbar. In beiden Fällen sollte das opt/<package>.txt File daraufhin geprüft werden, ob es die Korekten mode/gid/uid/flags optionen enthält bzw. Wenn das Script erst bem Booten erzeugt wurde, ein "chmod +x <Scriptname>" ausgeführt werden.
- Für temporäre Dateien sollte der Pfad /tmp genutzt werden. Es ist aber unbedingt darauf zu achten, daß hier nur wenig Platz ist weil dies in der Rootfs-Ramdisk liegt! Wenn mehr Platz benötigt wird, muss man sich eine eigene Ramdisk erstellen und mounten. Details dazu verrät der Abschnitt "RAM-Disks" dieser Dev-Doku.
- Damit temporäre Dateien eindeutige Namen erhalten sollte man grundsätzlich die aktuelle Prozess-ID an die Datei anhängen. Ein "guter" Dateiname ist also /tmp/<name-des-opts>-$$ statt /tmp/<name-des-opts>, wobei <name-des-opts> natürlich nicht so stehen bleiben soll, sondern ersetzt wird durch...was wohl?

8.7. Arbeit mit dem Paketfilter

8.7.1. Hinzufügen von eigenen Chains und Regeln

Zur Manipulation des Paketfilters steht eine Reihe von Routinen zur Verfügung, mit deren Hilfe man Chains und Regeln hinzufügen und wieder löschen kann.

add_chain chain, add_nat_chain chain Fügt eine Chain zur Filter- oder Nat-Tabelle hinzu.

flush_chain chain, flush_nat_chain chain Entfernt alle Regeln aus einer Chain

del_chain chain, del_nat_chain chain Entfernt eine Chain aus der Filter- oder Nat-Tabelle; Chains müssen leer sein, bevor sie gelöscht werden können und es darf auch keine Referenz mehr auf sie geben

add_rule, ins_rule, del_rule Fügt Regeln am Ende einer Chain (add) bzw. in die chain (ins) ein und löscht Regeln wieder (del). Ein Aufruf sieht wie folgt aus:

```
add_rule table chain rule comment
ins_rule table chain rule position comment
del_rule table chain rule comment
```

wobei die Parameter folgende Bedeutung haben:

table Die Tabelle, in deren Chain die Regel eingefügt werden soll

chain Die Chain, in die die Regel eingefügt werden soll

rule Die Regel, die eingefügt werden soll; das Format entspricht dem im Config-File verwendeten

position Die Position, an der die Regel eingefügt werden soll

comment Ein Kommentar, der zusammen mit der Regel angezeigt werden soll, wenn sich jemand den Paketfilter ansieht.

8.7.2. Einordnen in bestehende Regeln

Fli4l konfiguriert den Paketfilter mit einem gewissen Standardregelsatz. Will man eigene Regeln einfügen, wird man diese in der Regel nach dem Standardregelsatz einfügen wollen. Ebenfalls wird man wissen wollen, was denn die vom Nutzer gewünschte Aktion beim Verwerfen eines Paketes ist. Diese Informationen bekommt man für die FORWARD und INPUT Chain durch Aufruf zweier Funktionen wie im folgenden gezeigt:

```
get_defaults FORWARD # get default actions for forward chain
get_count    FORWARD # get number of default rules for forward
```

$res *res* enthält die Position nach den default rules

$drop *drop* enthält die Chain, in die verzweigt werden soll, wenn ein Paket verworfen wird

$reject enthält die Chain, in die verzweigt werden soll, wenn ein Paket abgelehnt wird

8.7.3. Erweiterung der Paketfilter-Matches

Fli4l verwendet in den Paketfilterregeln die Syntax `match:params`, um zusätzliche Bedingungen an die Pakete zu stellen (siehe mac:, limit:, length:, prot:, ...). Will man zusätzliche Matches hinzufügen, wird das folgendermassen gemacht:

1. Anlegen einer Datei opt/etc/rc.d/fwrules-<*name*>.ext. In diesem File steht in etwa folgendes:

   ```
   # extension is available
   foo_p=yes

   # the actual extension, adding matches to match_opt
   do_foo ()
   {
           param=$1
           get_negation $param
           match_opt="$match_opt -m foo $neg_opt --fooval $param"
   }
   ```

2. Testen der Erweiterung

   ```
   > cd opt/etc/rc.d
   > sh test-rules.sh 'foo:bar ACCEPT'
   add_rule filter FORWARD 'foo:bar ACCEPT'
   iptables -t filter -A FORWARD -m foo --fooval bar -s 0.0.0.0/0 \
       -d 0.0.0.0/0 -m comment --comment foo:bar ACCEPT -j ACCEPT
   ```

3. Aufnahme der Erweiterung und aller noch benötigten Dateien (iptables-Komponenten, netfilter-Kernelmodule) ins Archiv über einen der bekannten Mechanismen.

4. Zulassen der Erweiterung in der Konfiguration durch Erweitern von `PF_GENERIC_MATCH` in einem *.exp file, z.B. wie folgt:

```
+FW_GENERIC_MATCH='foo:bar' : ''
```

8.8. CGI-Erstellung für das httpd-Paket

8.8.1. Allgemeines zum Webserver

Der Webserver, der bei FLI4L verwendet wird, ist der `mini_httpd` von ACME Labs. Die Quellen und Manpages können unter http://www.acme.com/software/mini_httpd/ heruntergeladen werden. Allerdings wurden für FLI4L in der aktuellen Version (1.19) ein paar Änderungen vorgenommen, der Patch befindet sich im SRC-Paket im Verzeichnis `src/fli4l/httpd`.

8.8.2. Scriptnamen

Der Scriptname sollte möglichst vielsagend sein, damit er von anderen Scripts leichter zu unterscheiden ist und es keine Namensüberschneidungen bei verschiedenen OPTs gibt.

Damit die Scripts ausführbar gemacht werden und DOS-Zeilenumbrücke in UNIX-Zeilenumbrüche umgewandelt werden, muss in der opt/euer_opt.txt ein entsprechender Eintrag gemacht werden (s.o.).

8.8.3. Menü-Einträge

Um eine Eintrag im Menü vorzunehmen, muß eine Eintragung in der Datei `/etc/httpd/menu` vorgenommen werden. Dieser Mechanismus erlaubt es den OPTs, auch im laufendem Betrieb Änderungen am Menü vorzunehmen. Dies sollte nur mit dem Script `httpd-menu.sh` gemacht werden, die dieses darauf achtet, dass das Dateiformat dieser Datei immer schlüssig ist. Um neue Menüpunkte einzufügen, wird es folgendermaßen aufgerufen:

```
httpd-menu.sh add [-p <priority>] <link> <name> [section] [realm]
```

So wird ein Eintrag mit dem Namen <name> in den Abschnitt `[section]` eingetragen. Wenn `[section]` weggelassen wird, wird es standardmäßig in den Abschnitt „Opt" eingetragen. <link> gibt das Ziel des neuen Links an. <priority> spezifiziert die Priorität eines Menüeintrags in seinem Abschnitt. Wird sie nicht angegeben, wird die Standardpriorität 500 benutzt. Die Priorität sollte eine dreistellige Nummer sein. Je kleiner die Priorität, desto weiter oben steht der Link in dem Abschnitt. Soll ein Eintrag möglichst weit nach unten, so ist z.B. die Priorität 900 zu wählen. Bei gleicher Priorität werden die Einträge nach dem Ziel des Links sortiert. Bei `[realm]` wird der Breich angegeben, für den ein eingeloggter Benutzer mindestens die Berechtigung `view` haben muss, damit der Menüpunkt angezeigt wird. Wird `[realm]` nicht angegeben, wird der Menüpunkt immer angezeigt Siehe auch Abschnitt Benutzerrechte (Seite 336).

Beispiel:

```
httpd-menu.sh add "neuedatei.cgi" "Hier klicken" "Tools" "tools"
```

Dieses Beispiel erzeugt einen Link mit dem Titel „Hier klicken" und dem Link-Ziel „neuedatei.cgi" im Abschnitt „Tools" und legt diesen Abschnitt bei Bedarf an.

Das Skript kann auch Einträge aus dem Menü wieder entfernen:

```
httpd-menu.sh rem <link>
```

Mit diesem Aufruf wird der Eintrag mit dem Link <link> wieder entfernt. Achtung: Wenn mehrere Menüeinträge auf die gleiche Datei verweisen, werden alle diese Einträge aus dem Menü entfernt.

Da Abschnitte auch Prioritäten haben können, können diese auch manuell angelegt werden. Wird ein Abschnitt automatisch beim Hinzufügen eines Eintrages angelegt, erhält er automatisch die Priorität 500. Der Syntax zum Anlegen von Abschnitten lautet:

```
httpd-menu.sh addsec <priority> <name>
```

Auch hier sollte <priority> nur 3-stellige numerische Werte annehmen.

Wichtig: *Um sinnvolle Prioritäten in Erfahrung zu bringen, lohnt es sich, auf einem laufenden fli4l in die Datei* `/etc/httpd/menu` *zu schauen, die Prioritäten stehen in der 2. Spalte*

Der Vollständigkeit halber wird hier kurz auf das Dateiformat der Menüdatei eingegangen. Wem die Funktion von `httpd-menu.sh` reicht, der kann diesen Abschnitt überspringen. Die Datei `/etc/httpd/menu` hat den folgenden Aufbau: Sie ist in vier Spalten eingeteilt. In der ersten Spalte steht ein Kennbuchstabe, der Überschriften und Einträge unterscheidet. In der zweiten Spalte steht die Sortierungspriorität. Die dritte Spalte enthält bei Einträgen das Ziel des Links und bei Überschriften einen Bindestrich, da dieses Feld bei Überschriften keine Bedeutung hat. Im Rest der Zeile steht der Text, der nachher im Menü erscheint.

Überschriften benutzen den Kennbuchstaben 't', dann wird ein neuer Menüabschnitt begonnen. Normale Menüeinträge benutzen den Kennbuchstaben 'e'. Ein Beispiel:

```
t 300 - Mein tolles OPT
e 200 meinopt1.cgi Mach was tolles
e 500 meinopt1.cgi?mehr=ja Mach mehr tolles
```

Beim Bearbeiten dieser Datei muss man darauf achten, dass das `httpd-menu.sh`-Script die Datei immer sortiert abspeichert. Die einzelnen Abschnitte sind sortiert und die Einträge pro Abschnitt sind in diesem Abschnitt sortiert. Der genau Sortieralgorithmus kann aus `httpd-menu.sh` übernommen werden - besser wäre allerdings, dieses Script um mögliche neue Funktionen zu erweitern, damit alle Menü-Bearbeitungen an zentraler Stelle passieren.

8.8.4. Aufbau eines CGI-Skriptes

Die Kopfzeilen

Alle Skripte des Webservers sind einfache Shell-Skripte (Interpreter wie z.B. Perl, PHP, etc. sind viel zu groß für FLI4L). Sie sollten mit dem obligatorischen Script-Header anfangen (Verweis auf den Interpreter, Name, Sinn des Skriptes, Autor, Lizenz).

Das Hilfsskript cgi-helper

Gleich nach den Kopfzeilen sollte dann schon das Hilfsskript cgi-helper mit folgendem Aufruf eingebunden werden:

```
. /srv/www/include/cgi-helper
```

Wichtig ist ein Leerzeichen zwischen Punkt und Slash!

Dieses Skript stellt diverse Hilfsfunktionen bereit, die das Erstellen von CGIs für fli4l wesentlich vereinfachen sollen. Außerdem werden mit dem Einbinden des cgi-helper auch noch Standardaufgaben ausgeführt, wie beispielsweise das parsen von Variablen, die mit Formularen oder über die URL übergebenen wurden, oder das laden von Sprach- und CSS-Dateien.

Hier eine kleine Funktionsübersicht:

Name	Funktion
check_rights	Überprüfung der Benutzerrechte
http_header	Ausgabe eines Standard HTTP-Headers oder eines speziellen Headers, beispielsweise zum Download von Dateien
show_html_header	Ausgabe des kompletten Seitenheaders (inkl. HTTP-Header, Kopfzeile und Menü)
show_html_footer	Ausgabe des Abschlusses der HTML-Seite
show_tab_header	Ausgabe eines Inhalts-Fensters mit Tabs
show_tab_footer	Ausgabe des Abschlusses des Inhaltsfensters
show_error	Ausgabe eines Fensters für Fehlermeldungen (Farbe: rot)
show_info	Ausgabe eines Fensters für Informationen/ Erfolgsmeldungen (Farbe: grün)

Tabelle 8.3.: Funktionen des cgi-helper Skriptes.

Der Inhalt

Um ein einheitliches Design und vor allem die Kompatibilität mit zukünftigen fli4l-Versionen zu gewährleisten, ist es sehr zu empfehlen, die oben genannten Funktionen zu benutzen, auch wenn man in einem CGI theoretisch alle Ausgaben selbst generieren könnte.

Eine einfaches CGI-Skript könnte wie folgt aussehen:

```
#!/bin/sh
# --------------------
# Header (c) Autor Datum
# --------------------
# get main helper functions
. /srv/www/include/cgi-helper

 show_html_header "Mein erstes CGI"
 echo '   <h2>Willkommen</h2>'
 echo '   <h3>Dies ist ein Beispiel-CGI-Skript</h3>'
 show_html_footer
```

Der Parameter, welcher der Funktion show_html_header übergeben wird, wird dabei als Titel verwendet. Sie generiert automatisch das Menü und bindet ebenso automatisch zum Skript gehörende CSS und Sprachdateien ein. Voraussetzung dafür ist, dass diese sich in den Verzeichnissen /srv/www/css bzw. /srv/www/lang befinden und den selben Namen (aber natürlich eine andere Endung) wie das Skript haben. Ein Beispiel:

```
/srv/www/admin/OpenVPN.cgi
/srv/www/css/OpenVPN.css
/srv/www/lang/OpenVPN.de
```

Sowohl das Benutzen von Sprachdateien als auch von CSS-Dateien ist optional. Immer Eingebunden werden css/main.css und lang/main.xx wobei 'xx' der gewählten Sprache entspricht.

Der Funktion show_html_header können aber neben dem Titel noch weitere Parameter übergeben werden. Ein Aufruf mit allen möglichen Parametern könnte so aussehen:

```
show_html_header "Titel der Seite" "refresh=$time;url=$url;cssfile=$cssfile;showmenu=no
```

Alle zusätzlichen Parameter müssen, wie im Beispiel gezeigt, mit Anführungszeichen umschlossen und durch ein Semikolon getrennt werden. Eine Überprüfung der Syntax erfolgt NICHT! Es ist also notwendig sehr genau auf die Parameterübergabe zu achten.

Hier eine kurze Übersicht über die Funktion der Parameter:

- refresh=$time: Zeit in Sekunden in der die Seite vom Browser neu geladen werden soll.
- url=$url: Die URL die bei einem Refresh neu geladen wird.
- cssfile=$file: Name einer CSS-Datei, wenn diese vom Namen des CGIs abeicht.
- showmenu=no: Damit kann die Anzeige des Menüs und des Headers unterdrückt werden.

Sonstige Richtlinien zum Inhalt:

- Fasst euch kurz :-)
- Schreibt sauberes HTML (SelfHTML ist da ein guter Ansatzpunkt)
- Verzichtet auf hochmodernen Schnick-Schnack (JavaScript ist OK, wenn es nicht stört, sondern den Benutzer unterstützt, das ganze muß auch ohne JavaScript funktionieren)

Die show_tab_header Funktion

Damit der Inhalt eurer mithilfe des CGI erzeugten Webseite auch hübsch geordnet aussieht könnt ihr die cgi-helper Funktion show_tab_header nutzen. Sie erzeugt dann anklickbare 'Tabs', so dass ihr eure Seite in mehrere Bereiche unterteilen könnt, oder einfach nur eine Titelzeile (ein Tab).

Der show_tab_header Funktion werden Parameter immer in Paaren übergeben entgegen. Der erste Wert entspricht dem Titel eines Tabs, der 2. dem Link. Wird als Link die Zeichenkette 'no' übergeben, wird lediglich der Titel ausgegeben und der Tab ist nicht anklickbar (und blau).

Im folgenen Beispiel wir ein 'Fenster' mit dem Titel 'Ein tolles Fenster' erzeugt. Im Fenster steht 'foo bar':

```
show_tab_header "Ein tolles Fenster" "no"
echo "foo"
echo "bar"
show_tab_footer
```

In diesem Beispiel werden zwei anklickbare Auswahltabs generiert, die dem Skript die Variable action mit verschiedenen Werten übergibt.

```
show_tab_header "1. Auswahltab" "$myname?action=machdas" \
                "2. Auswahltab" "$myname?action=machdies"
echo "foo"
echo "bar"
show_tab_footer
```

Nun kann das Skript den Inhalt der Variablen FORM_action (siehe weiter unten zur Variablenauswertung) auswerten und je nach dem unterschiedliche Inhalte bereitstellen. Damit der angeklickte Tab auch ausgewählt erscheint und nicht mehr angeklickt werden kann, müsste der Funktion statt dem Link wie schon erwähnt ein 'no' übergeben werden. Das geht aber auch einfacher, wenn man sich an die Konvention in folgendem Beispiel hält:

```
_opt_machdas="1. Auswahltab"
_opt_machdies="2. Auswahltab"
show_tab_header "$_opt_machdas" "$myname?action=opt_machdas" \
                "$_opt_machdies" "$myname?action=opt_machdies"
case $FORM_action in
    opt_machdas) echo "foo" ;;
    opt_machdies) echo "bar" ;;
esac
show_tab_footer
```

Wird also dem Titel eine Variable übergeben, deren Namen dem Inhalt der Variablen action mit führendem '_' entspricht, wird der entsprechende Tab ausgewählt dargestellt.

Mehrsprachfähigkeit

Das Hilfsskript enthält auch alles, um CGI-Skripte mehrsprachfähig zu machen. Dazu müsst ihr 'nur' für alle Textausgaben Variablen mit führenden '_' verwenden und diese Variablen in den entsprechenden Sprachdateien definieren.

Beispiel:

lang/opt.de:

```
_opt_machdas="Eine Ausgabe"
```

lang/opt.en:

```
_opt_machdas="An Output"
```

admin/opt.cgi

```
...
echo $_opt_machdas
...
```

Formular-Auswertung

Um Formulare zu verarbeiten, muss man noch einige Dinge wissen. Egal ob die Formular-Methode GET oder POST verwendet wird, die Variablen finden sich nach dem Einbinden des cgi-helpers (welches wiederum das Hilfsprogramm proccgi aufruft) als shell-variable wieder, mit dem Prefix „FORM_". gespeichert. Wenn das Formularfeld also den Namen „eingabe" hatte, kann im Shell-Script mit `$FORM_eingabe` darauf zugegriffen werden.

Weitere Infos zum Programm proccgi: http://www.fpx.de/fp/Software/ProcCGI.html

8.8.5. Benutzerrechte

Um die Rechteverwaltung zu verwenden, müssen am Anfang des Scripts folgende Zeile eingefügt werden:

```
check_rights "misc" "test"
```

Dieses Script wird dann nur ausgeführt wenn der aktuelle Benutzer alle Rechte (`HTTPD_RIGHTS_x='all'`), alle Rechte in diesem Bereich (`misc`) (`HTTPD_RIGHTS_x='misc:all'`) oder das Recht hat, diese spezielle Aktion (`SEC_ACTION`) (`HTTPD_RIGHTS_x='misc:test'`) auszuführen.

8.8.6. Sonstiges

Dies und das (Ja, das ist auch noch wichtig!):

- Der `mini_httpd` schützt Unterverzeichnisse nicht mit Passwort. Es muß für jedes Verzeichniss eine eigene `.htaccess` angelegt werden oder ein Link auf eine andere htaccess-Datei angelegt werden.
- KISS - Keep it simple, stupid!
- Diese Angaben können sich jederzeit ohne Vorankündigung ändern!

8.8.7. Fehlersuche

Debug innerhalb des CGIs aktivieren, erstellt die Datei debug.log unter http://fli4l/admin/debug.log. Diese enthält alle Aufrufe des CGIs. Die Variabele ist nicht global, muss also in jeden Problem-CGI erneut gesetzt werden.

```
set_debug="yes"
. /srv/www/include/cgi-helper
```

Curl eignet sich hervoragend zur Fehlersuche, insbesondere wenn die HTTP Kopfzeilen nicht korrekt zusammengesetzt werden oder der Browser nur weiße Seiten anzeigt. Auch ist das Cachingverhalten moderner Webbrowser bei der Fehlersuche hinderlich.

Bespiel: mit curl den HTTP-Header dumpen (-D) und das CGI Output admin/mein.cgi betrachten. Dazu den Usernamen (-u) admin verwenden.

```
curl -D - http://fli4l/admin/mein.cgi -u admin
```

Mehr zu curl (Bezugsquellen und Informationen): http://de.wikipedia.org/wiki/CURL

8.9. Booten, Rebooten, Einwählen und Auflegen unter fli4l

8.9.1. Bootkonzept

fli4l 2.0 sollte eine saubere Installation auf eine Festplatte oder ein CompactFlash(TM)-Medium bieten, aber auch eine Installation auf ein Zip-Medium oder die Erstellung einer bootfähigen CD-ROM sollte möglich sein. Zusätzlich sollte die Festplattenversion sich nicht grundlegend von einer Installation von Diskette unterscheiden.

Diese Anforderungen wurden realisiert, indem die Dateien des opt.img aus der bisherigen RAM-Disk auf eine anderes Medium verlagert werden können. Dies kann eine Partition auf einer Festplatte bzw. einem CF-Medium sein. Dieses zweite Volume wird nach /opt gemountet und von dort werden nur noch symbolische Links in das rootfs angelegt. Das entstehende Layout im Root-Filesystem entspricht dann dem im opt-Verzeichnis der ausgepackten Fli4l-Distribution mit einer Ausnahme – das files-Prefix entfällt. Die Datei opt/etc/rc ist dann also direkt unter /etc/rc zu finden, opt/files/usr/local/bin/busybox unter /usr/local/bin/busybox. Daß diese Dateien unter Umständen nur Links auf ein read only gemountetes Medium sind, kann man ignorieren, solange man die Dateien nicht modifizieren möchte. Will man dies tun, muß man die Dateien vorher mit mk_writable schreibbar machen.

8.9.2. Start-Scripts

Scripts, die beim booten des Systems ausgeführt werden sollen, befinden sich unter opt/etc/{boot.d,rc.d} und werden auch in der Reihenfolge ausgeführt, also erst Dateien unter boot.d und danach unter rc.d. Da zum Ausführen dieser Scripte kein separater Prozess erzeugt wird, dürfen sie nicht mit “exit 0” abgeschlossen werden. Ein exit würde zum Abbruch des Bootvorganges führen.

Hilfsfunktionen für Startupscripte

In /etc/boot.d/base-helper werden verschiedene Funktionen bereitgestellt, die von den Startup-Scripten verwendet werden können. Das betrifft Dinge wie Debug-Support, Module-Behandlung, Fehler-Meldungen oder Ersetzung symbolischer Referenzen. Zum Laden der Module sollte aufgrund der automatischen Abhängikeitsauflösung do_modprobe verwendet werden. Sowohl mkfli4l als auch modprobe werten Modulabhängigkeiten aus und fügen benötgte Module zum Archiv hinzu bzw. laden sie beim Aufruf automatisch mit. Die do_insmod-Funktionen sind aus Kompatibilitätsgründen und für Spezialfälle weiterhin vorhanden.

Die einzelnen Funktionen werden im folgenden aufgelistet und kurz beschrieben.

begin_script <**symbol**> <**message**> Gibt die Message aus und aktiviert das Script debugging mittels *set -x*, falls <symbol>_DO_DEBUG auf yes steht.

end_script Gibt eine Abschluss-Message aus und stellt den ursprünglichen Debug-Zustand wieder her.

do_modrobe <**module**> <**params**> Laden das Moduls *module* mit den Parametern *params* bei gleichzeitiger Auflösung der Modulabhängigkeiten

do_modrobe_if_exists <**module_path**> <**module**> <**params**> Prüft, ob das Modul (/usr/lib/<kernel version>/<module path>/module) existiert und führt bei vorhandensein do_modprobe aus.

do_insmod <**module**> <**params**> Laden das Moduls *module* mit den Parametern *params*

do_insmod_once <**module**> <**params**> Wie do_insmod, nur das sichergestellt wird, daß das Module nur einmal geladen wird.

do_insmod_if_exists module_path <**module**> <**params**> Prüft, ob das Module existiert (/usr/lib/<kernel version>/<module path>/module) existiert und führt wenn es da ist, do_insmod aus.

do_insmod_if_exists_once <**module_path**> <**module**> <**params**> Prüft, ob das Module (/usr/lib/<kernel version>/<module path>/module) existiert und führt wenn es da ist und noch nicht geladen wurde, do_insmod aus.

mk_writable <**file**> Stellt sicher, daß die Datei *file* schreibbar ist. Befindet sich die Datei auf einem Read-Only Dateisystem und ist lediglich über einen Softlink ins Dateisystem eingebunden, wird eine lokale Kopie angelegt, die dann schreibbar ist.

unique <**elements**> Erzeugt aus den Parametern eine Liste, in der Elemente nur einmal auftauchen. Die Liste wird in der Variable *list* zurückgegeben.

log_error <**error message**> Schreibt die Fehlermeldung auf die Konsole und nach /boot_messages.txt. Wird keine Fehlermeldung als Parameter übergeben, liest log_error von stdin.

set_error <**error_message**> Gibt die Fehlermeldung aus und setzt ein Fehlerflag, das später via is_error geprüft werden kann.

is_error Setzt das Error-Flag zurück und liefert wahr zurück, falls es gesetzt war.

translate_ip_net <**value**> Ersetzt symbolische Referenzen in Parametern. Momentan finden folgende Übersetzungen statt:

ip adresse, netzwerk, none, default, pppoe, dynamic, Zahl Keine Übersetzung

any Wird nach 0.0.0.0/0 übersetzt

IP_NET_* wird durch das in der Konfiguration stehende Netzwerk ersetzt

IP_NET_*_IPADDR wird durch die in der Konfiguration stehende IP-Adresse ersetzt

IP_ROUTE_* wird durch die in der Konfiguration stehende Route ersetzt

@<**host name**> wird durch die in der Konfiguration für den angegeben Host spezifizierte IP-Adresse ersetzt

Nach dem Aufruf von translate_ip_net sollte man mit is_error prüfen, ob die Übersetzung korrekt erfolgt ist.

opt/etc/boot.d/rc*.*

Dateien, die in diesem Verzeichnis liegen, werden als erstes ausgeführt. Ihre Aufgabe ist es, das Boot-Device zu mounten, die rc.cfg einzulesen und das opt-Archive zu entpacken. Je nach Bootyp sind diese Scripte mehr oder weniger kompliziert und tun die folgenden Dinge:

- load drivers (optional)

- mount boot medium (optional)
- read rc.cfg
- extract/mount opt files (optional)

opt/etc/rc.d/rc*.*

Befehle, die immer beim Starten des Routers ausgeführt werden müssen, können in Scripts im Verzeichnis opt/etc/rc.d abgelegt werden. Hierbei gelten folgende Konventionen:

1. Es gilt folgende Namenskonvention: rc<dreistellige Zahl>.<Name des Opt's> Die Scripts werden in der Reihenfolge der Zahlen gestartet. Haben mehrere Scripts die gleiche Zahl, entscheidet die alphabetische Sortierung nach dem Punkt. Falls der Start eines Paketes erst nach einem anderen erfolgen darf, wird das durch die Zahl festgelegt.

   Hier eine ungefähre Richtlinie, welche Nummern für welche Aufgaben verwendet werden sollten:

Nummer	Aufgabe
000-099	Grundsystem (mounten, Zeitzone)
100-199	Kernel-Module (Treiber)
200-299	externe Verbindungen (PPPoE, I4L, PPtP)
300-399	Netzwerk (Routing, Interfaces, Paketfilter)
400-499	Server (DHCP, HTTPD, Proxy, etc.)
500-900	Freeform, use as you wish
900-997	Alles, was eine Einwahl hervorrufen könnte
998-999	reserved (don't use)

2. In diesen Scripts *müssen* alle Funktionen, die das rootfs verändern hinterlegt werden. (z.B. das Anlegen eines Verzeichnisses /var/log/lpd)

3. In diesen Scripts *dürfen* keine Schreibzugriffe auf den Dateien erfolgen, die Teil des opt-Archives sein können, da diese Dateien auf einem Read-Only-Medium liegen können. Muß man eine solche Datei modifizieren, muß man sie vorher via *mk_writable file* schreibbar machen. Durch den Aufruf der Funktion wird die Datei wenn nötig als schreibbare Kopie im rootfs abgelegt. Ist die Datei bereits schreibbar, bewirkt der *mk_writable*-Aufruf nichts. **Wichtig:** *mk_writable muß direkt auf den Files im rootfs angewendet werden, nicht über den Umweg /opt. Will man also /usr/local/bin/foo modifizieren, ruft man* mk_writable /usr/local/bin/foo *auf.*

4. Diese Scripte müssen vor der Ausführung der eigentlichen Befehle prüfen, ob das dazugehörige OPT auch aktiv ist. Das ist normalerweise durch eine einfache if-Abfrage erledigt:

```
if [ "$OPT_<OPT-Name>" = "yes" ]
then
    ...
    # Hier OPT starten!
    ...
fi
```

5. Um das debuggen zu erleichtern, sollten die Scripte mit *begin_script name message* und *end_script* geklammert werden:

```
if [ "$OPT_<OPT-Name>" = "yes" ]
then
    begin_script FOO ‘‘configuring foo ...’’
    ...
    end_script
fi
```

   Das debuggen einzelner Startupscripte kann man dann einfach via FOO_DO_DEBUG='yes' aktivieren.

6. Diese Scripte dürfen nicht direkt auf die Datei rc.cfg zugreifen. Dies ist auch nicht notwendig, weil alle Variablen aus dieser Datei direkt verfügbar sind bis der Startvorgang beendet wurde. Im Abschnitt "Umgang mit Konfigurationsvariablen" wurde erklärt, wie man Variablen für eine spätere Nutzung abspeichern kann.

7. Der Pfad /opt darf auch nicht als Speicherplatz für Datenbestände des Opts benutzt werden. Falls zusätzlicher Speicherplatz benötigt wird ist eine eigene Ramdisk zu erstellen oder die Angabe eines Pfades über eine Variable zu ermöglichen.

Stop-Scripts

Jeder Rechner muss mal heruntergefahren oder neu gestartet werden. Nun kann es vorkommen, dass man noch etwas durchführen muss, bevor der Rechner heruntergefahren oder neu gestartet wird. Zum herunterfahren und neu starten sind die Befehle halt und reboot zuständig. Diese Befehle werden auch aufgerufen, wenn man im IMONC auf die entsprechenden Buttons klickt.

Alle Stop-Scripts liegen im Verzeichnis opt/etc/rc0.d/. Ihre Dateinamen werden analog zu den Startscripten gebildet.

8.9.3. ttyI devices

Für die ttyI devices (/dev/ttyI0 ... /dev/ttyI15), über die die "Modem- Emulation" der ISDN-Karte genutzt werden kann, existiert ein Zähler, um Konflikte zwischen verschiedenen Paketen, die diese Devices nutzen, zu vermeiden. Hierzu wird beim Start des Routers die Datei /var/run/next_ttyI angelegt, die von den verschiedenen Addons abgefragt und fortgezählt werden muß. Mit dem folgenden Beispielscript kann dieser Wert abgefragt, um eins erhöht und wieder für das als nächstes startende Addon exportiert werden.

```
ttydev=‘cat /var/run/next_ttyI‘
if [ "$ttydev" -le 16 ]
then                                    # ttyI device available? yes
    ttydev=‘/usr/bin/expr $ttydev + 1‘  # ttyI device + 1
    echo $ttydev >/var/run/next_ttyI    # und wieder abspeichern
else                                    # ttyI device available? no
    /usr/local/bin/colecho "*** ERROR: no ttyI device for \
    <Name deines Addons> available ! ***" br x br
    sleep 10                            # wait, so user can read message
    ttydev_error=true                   # set error for later use
```

```
fi

if [ "$ttydev_error" != true ]          # start addon only if next tty device
                                        # was available to minimize error
then                                    # messages and minimize the
    ....                                # risk of uncomplete boot
fi
```

8.9.4. Scripte beim Einwählen und Auflegen

Nach dem Herstellen einer Dial-Up-Verbindung werden die Scripte unter /etc/ppp abgearbeitet. Hier können Opt-Pakete als Aktionen ausführen, die nach dem Herstellen der Verbindung bzw. nach dem Auflegen nötig sind. Benannt werden die Dateien wie folgt:

ip-up<dreistelligenummer>.<OPT-Name> Wird beim Verbinden ausgeführt

ip-down<dreistelligenummer>.<OPT-Name> Wird *nach* dem Trennen der Verbindung ausgeführt

In den ip-down-Scripts dürfen keine Aktionen ausgeführt werden, die zu einer erneuten Internet-Einwahl führen würden, da dadurch nur ein Dauer-Online-Zustand erreicht wird, was für Nicht-Flatrate-Benutzer ein teures Unterfangen ist.

Da für die einzelnen Scripte kein eigener Prozeß erzeugt wird, dürfen auch diese Scripts *nicht* mit “exit 0” abgeschlossen werden.

Hinweis: Wenn ein Script prüfen will ob überhaupt die ip-up Skripte ausgeführt werden kann es ab rc321 die Variable *”ip_up_events”* prüfen. Steht diese auf `"yes"` laufen die ip-up Skripte. Steht dort ein `"no"` laufen keine ip-up Skripte.

Variablen

Durch das spezielle Aufrufkonzept stehen alle Variablen aus der ip-up/ip-down auch in den einzelnen Scripts zur Verfügung. Folgende Variablen sind verfügbar:

$real_interface	Das aktuelle Interface, also z.B. ppp0, ippp0, ...
$interface	Das IMOND-Interface, also pppoe, ippp0, ...
$tty	Verbundenes Terminal... Möglicherweise leer!
$speed	Die Verbindunggeschwindigkeit, bei ISDN Z.B. 64000
$local	Die eigene IP-Adresse
$remote	Die IP-Adresse des Point-To-Point-Partners
$is_default_route	Gibt an, ob das aktuelle ip-up/down für das Interface durchgeführt wird, welches gerade die Default-Route hat (kann “yes” oder “no” sein)

Default-Route

Seit Version 2.1.0 werden die ip-up/down-Scripts nicht nur für das Default- Route-Interface gestartet, sondern für alle Verbindungen, die die ip-up/down- Scripts aufrufen. Um das alte Verhalten zu simulieren, muß in ip-up/down Scripts eine solche Abfrage eingefügt werden:

```
# is a default-route-interface going up?
if [ "$is_default_route" = "yes" ]
then
        # die eigentlichen Aktionen
fi
```

Natürlich darf das neue Verhalten auch für spezielle Aktionen ausgenutzt werden.

8.10. Paket TEMPLATE

Um einiges von dem hier beschriebene etwas zu veranschaulichen, liegt der fli4l-Distribution das Paket TEMPLATE bei. Dieses erklärt an kleinen Beispielen, wie:

- eine Konfigurationsdatei auszusehen hat (config/template.txt)
- eine check-Datei geschrieben wird (check/template.txt)
- die erweiterten Prüfmöglichekiten verwendet werden (check/template.ext)
- Konfigurationsvariablen für spätere Verwendung abgelegt werden können (opt/etc/rc.d/rc999.template)
- abgelegte Konfigurationsvariablen wieder ausgelesen werden (opt/files/usr/bin/template_show_config)

8.11. Aufbau der Boot-Diskette

Ab fli4l-Version 1.5 wird das Programm syslinux zum Booten von Diskette verwendet. Dieses hat den Vorteil, daß ein DOS-Filesystem auf der Diskette zur Verfügung steht.

Die Boot-Diskette enthält folgende Dateien:

ldlinux.sys	Bootloader syslinux
syslinux.cfg	Konfigurationsdatei für syslinux
kernel	Linux-Kernel
rootfs.img	Root-Filesystem
opt.img	Optionale Dateien: Treiber und Opt-Pakete
rc.cfg	Konfigurationsdatei mit den benutzten Variablen aus den Dateien des config-Verzeichnisses

Durch das Script mkfli4l.sh (bzw. mkfli4l.bat) werden zunächst die Dateien opt.img, syslinux.cfg und rc.cfg sowie das rootfs.img erzeugt. Die dafür nötige Dateiliste erstellt das Programm mkfli4l (im unix- bzw. windows-Unterverzeichnis). In dem bz2-Archiv (Mit Bzip2 komprimiertes Tar-Archiv) sind die benötigten Kernel- und Opt-Module enthalten. Früher war auch die Datei rc.cfg im opt-Archiv, diese liegt seit der fli4l-Version 2.0 direkt auf der Diskette.

Anschließend werden die Dateien kernel, rootfs.img, opt.tar.bz2 (opt.img) und rc.cfg zusammen mit den syslinux-Dateien auf die Diskette kopiert.

Beim Booten von fli4l wird über das Script /etc/rc die rc.cfg ausgewertet und das komprimierte Tar-Image in eine eigene Ramdisk oder auf eine schon existierende Partition extrahiert und diese nach /opt gemountet. Die Dateien opt/etc/* werden nun in das rootfs hineinkopiert und für die Dateien und Verzeichnisse in opt/* werden symbolische Links im rootfs erstellt. Danach werden die Scripte in /etc/rc.d in alphanumerischer Reihenfolge ausgeführt und somit die Treiber geladen und die Dienste gestartet.

8.12. Konfigurationsdateien

Hier werden die Dateien kurz aufgeführt, die vom fli4l-Router on-the-fly beim Booten erzeugt werden.

1. Konfiguration Provider
   - etc/ppp/pap-secrets
   - etc/ppp/chap-secrets
2. Konfiguration DNS
   - etc/resolv.conf
   - etc/dnsmasq.conf
   - etc/dnsmasq_dhcp.conf
   - etc/resolv.dnsmasq
3. Hosts-Datei
   - etc/hosts
4. imond-Konfiguration
   - etc/imond.conf

8.12.1. Konfiguration Provider

Für den ausgesuchten Provider wird in etc/ppp/pap-secrets die User-ID und das Password angepasst.

Beispiel für Provider Planet-Interkom:

```
# Secrets for authentication using PAP
# client        server  secret                  IP addresses
"anonymer"      *       "surfer"                *
```

Dabei ist im Beispiel “anonymer” die USER-ID. Als Remote-Server wird prinzipiell jeder erlaubt (deshalb “*”). “surfer” ist das Password für den Provider Planet-Interkom.

8.12.2. Konfiguration DNS

Man kann den fli4l-Router als DNS-Server einsetzen. Warum dies sinnvoll - ja sogar bei Windows-Rechnern im LAN zwingend notwendig ist, wurde bereits in ../doc erläutert.

Die Resolver-Datei etc/resolv.conf enthält den Domainnamen und den zu verwendenten Nameserver. Sie hat folgenden Inhalt (wobei domain.de nur ein Platzhalter für den Wert der Konfigurationsvariable `DOMAIN_NAME` ist):

```
search domain.de
nameserver 127.0.0.1
```

Der Nameserver dnsmasq wird über die Datein etc/dnsmasq.conf konfiguriert. Sie wird beim Booten vom Script rc001.base sowie rc370.dnsmasq automatisch erzeugt und könnte wie folgt aussehen:

```
user=dns
group=dns
resolv-file=/etc/resolv.dnsmasq
no-poll
no-negcache
bogus-priv
log-queries
domain-suffix=lan.fli4l
local=/lan.fli4l/
domain-needed
expand-hosts
filterwin2k
conf-file=/etc/dnsmasq_dhcp.conf
```

8.12.3. Hosts-Datei

Sollte jedem bekannt sein :-) Ist eigentlich überflüssig, wenn zusätzlich ein lokaler DNS-Server gestartet wird.

8.12.4. imond-Konfiguration

Die Datei etc/imond.conf wird u.a. aus den Konfigurationsvariablen `CIRC_x_NAME`, `CIRC_x_ROUTE`, `CIRC_x_CHARGEINT` und `CIRC_x_TIMES` konstruiert. Sie kann aus bis zu 32 Zeilen (ohne die Kommentarzeilen) bestehen. Jede Zeile besteht aus 8 Spalten:

1. Bereich Wochentag bis Wochentag
2. Bereich Stunde bis Stunde
3. Device (ipppX oder isdnX)
4. Circuit mit Default-Route: yes/no
5. Telefonnummer
6. Name des Circuits
7. Telefonkosten pro Minute in DM
8. Zeittakt (charge interval) in Sekunden

Hier ein Beispiel:

```
#day  hour  device  defroute  phone        name        charge  ch-int
Mo-Fr 18-09 ippp0   yes       010280192306 Addcom      0.0248  60
Sa-Su 00-24 ippp0   yes       010280192306 Addcom      0.0248  60
Mo-Fr 09-18 ippp1   yes       019160       Compuserve  0.019   180
Mo-Fr 09-18 isdn2   no        0221xxxxxxx  Firma       0.08    90
Mo-Fr 18-09 isdn2   no        0221xxxxxxx  Firma       0.03    90
Sa-Su 00-24 isdn2   no        0221xxxxxxx  Firma       0.03    90
```

Weitere Erklärungen zum Least-Cost-Routing findet man in ../doc.

A. Anhang zum Basispaket

A.1. fli4l-Diskette auf Darwin oder Apple Mac OS X erzeugen

Vorsicht: Diese Funktionalität ist experimentell und noch nicht sehr ausgiebig getestet. Unerfahrene Benutzer sollten ihre Diskette nicht auf Mac OS X erzeugen.

A.1.1. Voraussetzungen

Es wird folgende Software benötigt um die Diskette zu erzeugen:

Apple Xcode	http://developer.apple.com/technologies/tools/features.html	Enthält den benötigten Compiler und zusätzliche Programme, Registrierung nötig
mtools	http://www.gnu.org/software/mtools/intro.html	mtools wird benötigt um die Dateien auf die Diskette zu kopieren. Vielleicht kann man sie auch über Fink (http://www.finkproject.org/) installieren.

A.1.2. Einleitung

Im Moment ist noch viel Arbeit mit Terminal.app nötig, um die fli4l-Diskette auf Apple Mac OS X zu erzeugen. Wer kein Terminal bedienen kann oder will muss warten bis vielleicht eine grafische Oberfläche verfügbar ist.

Für die folgenden Abschnitte wird erwartet, dass ein Terminalfenster (/Applications/Utilities/Terminal.app) geöffnet ist.

A.1.3. fli4l-Distribution entpacken

Die heruntergeladenen Pakete können mit den folgenden Befehlen entpackt werden:

```
cd ~/Documents/
tar xvzf ${Pfad-zu-fli4l}/3.6.2.tar.gz
```

A.1.4. mtools manuell kompilieren

Damit die mtools kompiliert werden können muss zuerst Apple Xcode (http://developer.apple.com/xcode/) installiert sein.

Zuerst muss der Quellcode der mtools von http://www.gnu.org/software/mtools/intro.html heruntergeladen werden. Zum Zeitpunkt, als diese Doku geschrieben wurde, war Version 3.9.9 aktuell. Diese können zum Beispiel nach ~/Documents/ verschoben werden. Im Terminal können sie dann mit folgenden Befehlen entpackt werden:

```
cd ~/Documents/
tar xvzf mtools-3.9.9.tar.gz
```

Danach müssen die Quellen konfiguriert werden:

```
cd mtools-3.9.9/
./configure
```

Und kompiliert werden sie mit dem Befehl “make”:

```
make
```

Dann müssen die mtools installiert werden. Dies geschieht mit dem Befehl

```
sudo make install
```

Dies setzt voraus, dass der aktuelle Benutzer ein Administrator ist. Wenn der root-Benutzer aktiviert ist, so kann auch “su” verwendet werden.

A.1.5. Diskette erzeugen

Die Diskette kann im fli4l-Verzeichnis mit dem Befehl ./mkfloppy-darwin.sh erzeugt werden. Das Skript fragt dann automatisch, welches Diskettenlaufwerk verwendet werden soll, sofern mehrere vorhanden sind.

A.2. Tokenring – Auszug aus Configure.help des Linux-Kernels

```
Token Ring driver support
CONFIG_TR
  Token Ring is IBM's way of communication on a local network; the
  rest of the world uses Ethernet. To participate on a Token Ring
  network, you need a special Token ring network card. If you are
  connected to such a Token Ring network and want to use your Token
  Ring card under Linux, say Y here and to the driver for your
  particular card below and read the Token-Ring mini-HOWTO, available
  from <http://www.tldp.org/docs.html#howto>. Most people can
  say N here.

IBM Tropic chipset based adapter support
CONFIG_IBMTR
  This is support for all IBM Token Ring cards that don't use DMA. If
  you have such a beast, say Y and read the Token-Ring mini-HOWTO,
  available from <http://www.tldp.org/docs.html#howto>.

  Warning: this driver will almost definitely fail if more than one
  active Token Ring card is present.

  This driver is also available as a module ( = code which can be
```

```
  inserted in and removed from the running kernel whenever you want).
  The module will be called ibmtr.o. If you want to compile it as a
  module, say M here and read <file:Documentation/modules.txt>.

IBM Olympic chipset PCI adapter support
CONFIG_IBMOL
  This is support for all non-Lanstreamer IBM PCI Token Ring Cards.
  Specifically this is all IBM PCI, PCI Wake On Lan, PCI II, PCI II
  Wake On Lan, and PCI 100/16/4 adapters.

  If you have such an adapter, say Y and read the Token-Ring
  mini-HOWTO, available from <http://www.tldp.org/docs.html#howto>.

  This driver is also available as a module ( = code which can be
  inserted in and removed from the running kernel whenever you want).
  The module will be called olympic.o. If you want to compile it
  as a module, say M here and read <file:Documentation/modules.txt>.

  Also read <file:Documentation/networking/olympic.txt> or check the
  Linux Token Ring Project site for the latest information at
  <http://www.linuxtr.net/>.

IBM Lanstreamer chipset PCI adapter support
CONFIG_IBMLS
  This is support for IBM Lanstreamer PCI Token Ring Cards.

  If you have such an adapter, say Y and read the Token-Ring
  mini-HOWTO, available from <http://www.tldp.org/docs.html#howto>.

  This driver is also available as a modules ( = code which can be
  inserted in and removed from the running kernel whenever you want).
  The modules will be called lanstreamer.o. If you want to compile it
  as a module, say M here and read <file:Documentation/modules.txt>.
Generic TMS380 Token Ring ISA/PCI/MCA/EISA adapter support
CONFIG_TMS380TR
  This driver provides generic support for token ring adapters
  based on the Texas Instruments TMS380 series chipsets.  This
  includes the SysKonnect TR4/16(+) ISA (SK-4190), SysKonnect
  TR4/16(+) PCI (SK-4590), SysKonnect TR4/16 PCI (SK-4591),
  Compaq 4/16 PCI, Thomas-Conrad TC4048 4/16 PCI, and several
  Madge adapters.  If you say Y here, you will be asked to select
  which cards to support below.  If you're using modules, each
  class of card will be supported by a separate module.

  If you have such an adapter and would like to use it, say Y and
  read the Token-Ring mini-HOWTO, available from
  <http://www.tldp.org/docs.html#howto>.
```

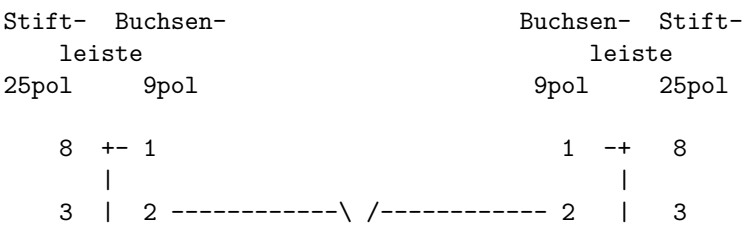
```
  Also read the file <file:Documentation/networking/tms380tr.txt> or
  check <http://www.auk.cx/tms380tr/>.

  This driver is also available as a module ( = code which can be
  inserted in and removed from the running kernel whenever you want).
  The module will be called tms380tr.o. If you want to compile it
  as a module, say M here and read <file:Documentation/modules.txt>.

Generic TMS380 PCI support
CONFIG_TMSPCI
  This tms380 module supports generic TMS380-based PCI cards.

  These cards are known to work:
     - Compaq 4/16 TR PCI
     - SysKonnect TR4/16 PCI (SK-4590/SK-4591)
     - Thomas-Conrad TC4048 PCI 4/16
     - 3Com Token Link Velocity

  This driver is available as a module ( = code which can be
  inserted in and removed from the running kernel whenever you want).
  The module will be called tmspci.o. If you want to compile it
  as a module, say M here and read <file:Documentation/modules.txt>.

Generic TMS380 ISA support
CONFIG_TMSISA
  This tms380 module supports generic TMS380-based ISA cards.

  These cards are known to work:
     - SysKonnect TR4/16 ISA (SK-4190)

  This driver is available as a module ( = code which can be
  inserted in and removed from the running kernel whenever you want).
  The module will be called tmsisa.o. If you want to compile it
  as a module, say M here and read <file:Documentation/modules.txt>.

Madge Smart 16/4 PCI Mk2 support
CONFIG_ABYSS
  This tms380 module supports the Madge Smart 16/4 PCI Mk2
  cards (51-02).

  This driver is available as a module ( = code which can be
  inserted in and removed from the running kernel whenever you want).
  The module will be called abyss.o. If you want to compile it
  as a module, say M here and read <file:Documentation/modules.txt>.

SMC ISA/MCA Token Ring adapter support
```

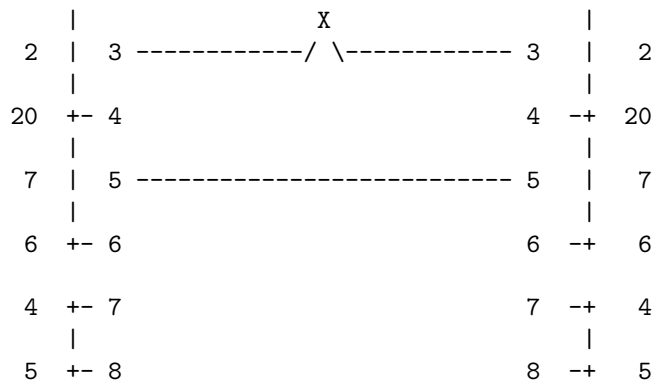
```
CONFIG_SMCTR
  This is support for the ISA and MCA SMC Token Ring cards,
  specifically SMC TokenCard Elite (8115T) and SMC TokenCard Elite/A
  (8115T/A) adapters.

  If you have such an adapter and would like to use it, say Y or M and
  read the Token-Ring mini-HOWTO, available from
  <http://www.tldp.org/docs.html#howto> and the file
  <file:Documentation/networking/smctr.txt>.

  This driver is also available as a module ( = code which can be
  inserted in and removed from the running kernel whenever you want).
  The module will be called smctr.o. If you want to compile it
  as a module, say M here and read <file:Documentation/modules.txt>.

3COM 3C359 Token Link Velocity XL PCI adapter support
CONFIG_3C359
  This is support for the 3Com PCI Velocity XL cards, specifically
  the 3Com 3C359, please note this is not for the 3C339 cards, you
  should use the tms380 driver instead.

  If you have such an adapter, say Y and read the Token-Ring
  mini-HOWTO, available from <http://www.tldp.org/docs.html#howto>.

  This driver is also available as a module ( = code which can be
  inserted in and removed from the running kernel whenever you want).
  The module will will be called 3c359.o. If you want to compile it
  as a module, say M here and read Documentation/modules.txt.

  Also read the file <file:Documentation/networking/3c359.txt> or check the
  Linux Token Ring Project site for the latest information at
  <http://www.linuxtr.net>
```

A.3. Nullmodemkabel

Für die Verwendung des optionalen Programmpakets PPP (Seite 211) benötigt man ein Nullmodemkabel.

Dieses muß mindestens 3 Adern haben. Hier die Anschluß-Belegung:





Bei den Steckern müssen die im Schaltbild gezeigte Brücken eingelötet werden.

A.4. Serielle Console

fli4l kann ohne Monitor und Tastatur eingesetzt werden. Ein Nachteil davon ist, daß eventuelle Fehlermeldungen nicht bemerkt werden, weil sich nicht alle Meldungen über die syslog-Schnittstelle umleiten lassen.

Eine Möglichkeit ist die Umlenkung der Console-Meldungen auf seinen PC oder auf ein klassisches Terminal, nämlich über die serielle Schnittstelle. Die Konfiguration erfolgt ueber die Variablen SER_CONSOLE (Seite 30), SER_CONSOLE_IF (Seite 30) und SER_CONSOLE_RATE (Seite 30).

Rechner mit älteren Mainboards/Karten unterstützen keine höheren Geschwindigkeiten als 38400 Bd. Deshalb sollte man es zunächst mit höchstens 38400 Bd probieren, bevor man sich an höhere Geschwindigkeiten heranwagt. Da lediglich Text-Ausgaben über die Console gehen, sind höhere Geschwindigkeiten eigentlich auch gar nicht notwendig.

Sämtliche Meldungen, die normalerweise auf der Console ausgegeben werden, werden nun auf die serielle Schnittstelle gelenkt – auch die Meldungen des Bootvorgangs!

Als Kabel zum Terminal oder PC mit Terminalemulation kommt ein Nullmodemkabel (Seite 349) zum Einsatz. Ich rate aber davon ab, ein Standard-Nullmodemkabel zu verwenden, weil dort normalerweise auch die Handshake-Leitungen verdrahtet sind. Ist das Terminal bzw. der PC abgeschaltet (oder die Terminalemulation nicht geladen), kann es bei Verwendung eines Standard-Nullmodemkabels zu einem Hangup von fli4l kommen!

Deshalb ist hier eine spezielle Verdrahtung notwendig, um fli4l auch mit abgeschaltetem Terminal betreiben zu können. Es wird dafür ein 3-adriges Kabel benötigt, wobei einige Kontakte an den Steckern gebrückt werden müssen. Siehe dazu bei Nullmodemkabel (Seite 349).

A.5. Programme

Um Platz auf der Diskette zu sparen, wird unter anderem das Paket “BusyBox” verwendet. Das Programm ist ein einzelnes Exceutable, welches die Standard-Unix-Programme

```
[, ash, basename, bunzip2, busybox, bzcat, cat, chgrp, chmod,
chown, chroot, cp, cut, date, dd, df, dirname, echo, expr, false,
grep, gunzip, gzip, halt, hostname, init, insmod, ip,
```

```
kill, killall, ln, logger, ls, mkdir, mkfs.minix, mknod, mount,
mv, nslookup, ping, pipe_progress, poweroff, ps, pwd, reboot, rm,
rmmod, sed, seq, sh, sleep, sort, sync, tar, test, tftp,
tr, true, tty, umount, zcat
```

nachbildet. Zumeist sind es jedoch "minimalistische" Implementationen, welche nicht den vollen Funktionsumfang abdecken, aber vollauf den bescheidenen Anforderungen von fli4l genügen.

BusyBox steht unter GPL und ist als Source komplett erhältlich unter
http://www.busybox.net/

A.6. Andere i4l-Tools

Es gibt für isdn4linux viele weitere Tools, die auch fli4l bereichern würden. Das Problem ist leider der Platz! Bestimmt wäre isdnlog als Tool zum Berechnen der Online-Gebühren wesentlich geeigneter, jedoch ist isdnlog einfach zu fett!

imond braucht weniger als 10% des Platzes, übernimmt dabei Monitoring, Controlling und LC-Routing, wenn auch nicht alles ganz perfekt ist.

A.7. Fehlersuche

Hilfreich bei der Fehlersuche sind natürlich einmal die Console-Outputs. Diese rauschen aber oft einfach so durch, dass man gar nicht mehr mitlesen kann. Hinweis: Mit SHIFT-BILD-RAUF kann man zurück-, mit SHIFT-BILD-RUNTER wieder vorblättern.

Falls im Betrieb des Routers Fehlermeldungen "try-to-free-pages" auftreten ist zuwenig für Programme nutzbares RAM übrig. Als Abhilfe stehen dann folgende Optionen zur Verfügung:

- mehr RAM einbauen
- weniger Opt-Pakete einsetzen
- eine Festplatteninstallation nach Typ B (Seite 16) durchführen

Auch proc-Dateien können bei der Fehlersuche helfen. z.B. gibt der Befehl

```
cat /proc/interrupts
```

die von den Treibern verwendeten Interrupts aus – nicht die tatsächlich von der Hardware belegten!

Weitere interessante Dateien unter /proc sind devices, dma, ioports, kmsg, meminfo, modules, uptime, version und pci (falls der Router einen PCI-Bus hat).

Meist liegt ein Verbindungsproblem bei ipppd vor, insbesondere bei der Authentifizierung. Dann helfen oft die Variablen

```
OPT_SYSLOGD='yes'

OPT_KLOGD='yes'
```

in config/base.txt und

```
ISDN_CIRC_x_DEBUG='yes'
```

in config/isdn.txt weiter.

A.8. Literaturhinweise

- Computer Networks, Andy Tanenbaum
- TCP/IP Netzanbindung von PCs, Craig Hunt
- TCP/IP, Kevin Washburn, Jim Evans, Verlag: Addison-Wesley, ISBN: 3-8273-1145-4
- TCP/IP Netzanbindung von PCs, ISBN 3-930673-28-2
- TCP/IP Netzwerk Administration, ISBN 3-89721-110-6
- Linux-Anwenderhandbuch, ISBN 3-929764-06-7
- TCP/IP im Detail: http://www.nickles.de/c/s/ip-adressen-112-1.htm
- Generell das online Linuxanwenderhandbuch von Lunetix unter: http://www.linux-ag.com/LHB/
- Einführung in die Linux-Firewall: http://www.little-idiot.de/firewall/

A.9. Präfixe

Bei den Einheiten richten die Präfixe in dieser Doku sich nach `IEC 60027-2`.
Siehe: http://physics.nist.gov/cuu/Units/binary.html.

A.10. Gewähr und Haftung

Natürlich kann für das gesamte fli4l-Paket oder für Teile davon keine Gewähr dafür übernommen werden, dass es überhaupt funktioniert oder dass irgendeine Dokumentation in diesem Verzeichnis oder einem der Unterverzeichnisse korrekt ist.

Auch ist jede Haftung wegen evtl. entstandender Schäden oder Kosten ausgeschlossen!

A.11. Danke

In diesem Abschnitt der Dokumentation soll all dennen durch namentlich Nennung gedankt werden die zur Entwicklung von fli4l beitragen bzw. beigetragen haben.

A.11.1. Projektgründung

Meyer, Frank E-Mail: `frank(at)fli4l(dot)de`

Frank startete am 04.05.2000 das Projekt fli4l!
Siehe: http://www.fli4l.de/home/eigenschaften/historie/

A.11.2. Entwickler- und Testteam

Das fli4l-Team bilden (in alphabetischer Reihenfolge):

Bork, Thomas (*lpdsrv*)
E-Mail: knuffelmail(at)gmx(dot)de
Behrends, Arno (*Support*)
E-Mail: arno(at)fli4l(dot)de
Eckhofer, Felix (*Dokumentation, Howtos*)
E-Mail: felix(at)fli4l(dot)de
Gruetzmacher, Tobias (*Mini-httpd, imond, proxy*)
E-Mail: tobias(at)portfolio16(dot)de
Hilbrecht, Claas (*VPN, Kernel*)
E-Mail: claas(at)jucs-kramkiste(dot)de
Horsmann, Karsten (*Mini-httpd, WLAN*)
E-Mail: fli4l(at)netzwech(dot)de
Knipping, Michael (*Accounting*)
E-Mail: fli4l(at)knibo(dot)de
Krister, Stefan (*Opt-Cop, lcd4linux*)
E-Mail: stefan(dot)krister(at)chreativ(dot)chaos(dot)de
Miksch, Gernot (*LCD*)
E-Mail: gernot_miksch(at)gmx(dot)de
Resch, Robert (*PCMCIA, WLAN*)
E-Mail: rresch(at)gmx(dot)de
Schiefer, Peter (*fli4l-CD, Opt-Cop, Webseite, Releasemanagement*)
E-Mail: peter(at)fli4l(dot)de
Schulz, Christoph (*Dokumentation, IPv6*)
E-Mail: fli4l(at)kristov(dot)de
Vosselman, Arwin (*LZS-Kompression, Dokumentation*)
E-Mail: arwin(at)xs4all(dot)nl
Wallmeier, Nico (*Windows-Imonc*)
E-Mail: nico(at)fli4l(dot)de
Walter, Gerd (*UMTS*)
E-Mail: fli4l(at)hgwb(dot)de
Walter, Oliver (*QoS*)
E-Mail: owb(at)gmx(dot)de
Weiler, Manuela (*CD-Versand, Kassenwart*)
E-Mail: -
Weiler, Marcel (*Qualitätsmanagement*)
E-Mail: -
Wolter, Jean (*Paketfilter, uClibc*)
E-Mail: -

Das fli4l-Test- und Übersetzerteam bilden (in alphabetischer Reihenfolge):

Charrier, Bernard
Fischer, Joerg
Frauenhoff, Peter
Schliesing, Manfred
Schlund, Christian
Schultz, Michael
Vogel, Samuel
Zenke, Michael

A.11.3. Entwickler- und Testteam (nicht mehr aktive)

Arndt, Kai-Christian (*USB*)
Bauer, Jürgen (*LCD-Package, fliwiz*)
Blokland, Kees (*Englische Übersetzung*)
Cerny, Carsten (*Webseite, fliwiz*)
Dawid, Oliver (*dhcp, uClibc*)
Ebner, Hannes (*QoS*)
Grabner, Hans-Joerg (*imonc*)
Grammel, Matthias (*Englische Übersetzung*)
Hahn, Joerg (*IPSEC*)
Hanselmann, Michael (*Mac OS X/Darwin*)
Hoh, Jörg (*Newsletter, NIC-DB, Veranstaltungen*)
Hornung, Nicole (*Verein*)
Janus, Frank (*LCD*)
Kaiser, Gerrit (*Logo*)
Karner, Christian (*PPTP-Package*)
Klein, Marcus (*Problemfeedback*)
Lammert, Gerrit (*HTML-Dokumentation*)
Lanz, Ulf (*LCD*)
Lichtenfeld, Nils (*QoS*)
Neis, Georg (*fli4l-CD, Dokumentation*)
Peiser, Steffen (*FAQ*)
Peus, Christoph (*uClibc*)
Pohlmann, Thorsten (*Mini-httpd*)
Raschel, Tom (*IPX*)
Reinard, Louis (*CompactFlash*)
Schäfer, Harald (*HDD-Support*)
Strigler, Stefan (*GTK-Imonc, Opt-DB, NG*)
Zierer, Florian (*Wunschliste*)

A.11.4. Sponsoren

Auch ist mittlerweile fli4l als Wort-/Bildmarke eingetragen. Folgende fli4l-Anwender (neben einigen, die nicht genannt werden wollten) haben geholfen das dafür nötige Geld zusammen zu bekommen:

Bebensee, Norbert
Becker, Heiko
Behrends, Arno
Böhm, Stefan
Brederlow, Ralf
Groot, Vincent de
Hahn, Olaf
Hogrefe, Paul
Holpert, Christian
Hornung, Nicole
Kuhn, Robert
Lehnen, Jens
Ludwig, Klaus-Ruediger
Mac Nelly, Christa
Mahnke, Hans-Jürgen
Menck, Owen
Mende, Stefan
Mücke, Michael
Roessler, Ingo
Schiele, Michael
Schneider, Juergen
Schönleber, Suitbert
Sennewald, Matthias
Sternberg, Christoph
Vollmar, Thomas
Walter, Oliver
Wiebel, Christian
Woelk, Fabian

Seit einiger Zeit hat fli4l nun auch seine eigenen Sponsoren, die mit Ihrer (Hardware-)Spende die Weiterentwicklung von fli4l unterstützen. Dabei handelt es sich um Adapter, CompactFlash und Ethernetkarten.

Hardwarespender (in alphabetischer Reihenfolge):

Baglatzis, Stephanos
Bauer, Jürgen
Dross, Heiko
Kappenhagen, Wenzel
Kipka, Joachim
Klopfer, Tom
Peiser, Steffen
Reichelt, Detlef
Reinard, Louis
Stärkel, Christopher

Weitere Sponsoren sind auf der fli4-Homepage gelistet:
http://www.fli4l.de/sonstiges/sponsoren/

A.12. Feedback

Kritik, Feedback und Zusammenarbeit ist jederzeit willkommen.

Die primäre Anlaufstelle dafür sind die fli4l-Newsgroups. Wer Probleme bei der Einrichtung eines fli4l-Routers hat, sollte sich erst FAQ, Howtos und NG-Archiv anschauen, bevor er sich an die Newsgroups wendet. Informationen über die verschiedenen Gruppen und die Netiquette findet man auf der fli4l-Webseite:

http://www.fli4l.de/hilfe/newsgruppen/
http://www.fli4l.de/hilfe/faq/
http://www.fli4l.de/hilfe/howtos/

Gerade weil für fli4l-Router meist ältere Hardware eingesetzt wird, kann es immer wieder mal zu Problemen kommen. Informationen können anderen fli4l-Usern bei Problemen mit der Hardware weiterhelfen, denn es gibt immer wieder Probleme mit den PC-Karten bzgl. I/O-Adressen, Interrupts und so weiter.

Auf der fli4l-Webseite wurde eine Netzwerkkarten-Datenbank eingerichtet, in die man z.B. die passenden Treiber für eine bestimmte Karte eintragen kann.

http://www.fli4l.de/hilfe/nic-db/

Viel Spaß mit fli4l!

B. Anhänge der optionalen Pakete

B.1. CHRONY - Benachrichtigung anderer Applikationen über Timewarps

Stellt chrony fest, dass die Uhr sehr weit von der tatsächlichen Uhrzeit abweicht, korrigiert chrony die Zeit in einem grossen Schritt und führt Scripts aus, um andere Anwednungen von diesem Zeitsprung zu informieren. Um z.B. den Imond von einem Zeitsprung zu informieren, macht chrony folgendes:

1. Scripte ins Archiv aufnehmen

   Chrony fügt dem Archiv zwei Files hinzu:

   ```
   start_imond  yes  etc/chrony.d/timewarp.sh mode=555 flags=sh
   start_imond  yes  etc/chrony.d/timewarp100.imond mode=555 flags=sh
   ```

   timewarp.sh führt alle Scripts im gleichen Verzeichnis aus, die dem Namen timewarp<3 ziffern>.<name> entsprechen.

2. Script zur Verfügung stellen

   chrony nimmt folgendes Script mit ins Archiv auf:

   ```
   # inform imond about time warp
   imond-stat "adjust-time $timewarp 1"
   ```

   Damit wird der imond über den Zeitsprung informiert und kann seine interne Zeitbasis anpassen.

B.2. DSL - PPPD und Active Filter

Für fli4l setzen wir den im Link angegebenen Ausdruck ein:

```
'outbound and not icmp[0] != 8 and not tcp[13] & 4 != 0'
```

und erreichen damit, daß grundsätzlich nur vom lokalen Netz ins Internet gesendete Pakete die Verbindung offen halten, mit ein paar Ausnahmen:

- *TCP-RST*: Antworten auf abgelehnte Verbindungswünsche von außen setzen den Timeout nicht zurück,
- *ICMP*: gesendete ICMP-Nachrichten setzen den Timeout ebenfalls nicht zurück, es sei denn, es wird ein Echo-Request gesendet.

Dieser Ausdruck wird normalerweise vom PPPD in einen vom Kernel verwendbaren Paket-Filter umgesetzt. Dieser würde in diesem Beispiel wir folgt aussehen:

```
#
# Expression: outbound and not icmp[0] != 8 and not tcp[13] & 4 != 0
#
(000) ldb      [0]
(001) jeq      #0x0            jt 17   jf 2
(002) ldh      [2]
(003) jeq      #0x21           jt 4    jf 18
(004) ldb      [13]
(005) jeq      #0x1            jt 6    jf 11
(006) ldh      [10]
(007) jset     #0x1fff         jt 18   jf 8
(008) ldxb     4*([4]&0xf)
(009) ldb      [x + 4]
(010) jeq      #0x8            jt 18   jf 17
(011) jeq      #0x6            jt 12   jf 18
(012) ldh      [10]
(013) jset     #0x1fff         jt 18   jf 14
(014) ldxb     4*([4]&0xf)
(015) ldb      [x + 17]
(016) jset     #0x4            jt 17   jf 18
(017) ret      #0
(018) ret      #4
```

Dieser Code muß generiert werden und dazu enthält der pppd normalerweise einen kleinen Compiler, der Ausdrücke in den entsprechenden Code für den Paketfilter des Kerns umwandelt. Dieser Code würde denn ppp-Daemon jedoch um 100Kbyte vergrößern und damit sinnlos Platz auf der Floppy verschwenden. Deshalb wurde beschlossen, den Compiler aus dem pppd rauszunehmen und separat zur Verfügung zu stellen (pfc). Der pppd liest dann lediglich den offline compilierten Code und übergibt ihn an den Kern. Der vorcompilierte Filterausdruck liegt unter opt/etc/ppp/filter.

Will man den von fli4l verwendeten Ausdruck verändern, muß man den Paketfiltercode neu generieren. Dazu benötigt man den packet filter compiler. Zum Compilieren setzt man entweder OPT_PFC (Seite 118) auf yes und compiliert den Ausdruck auf dem Router oder man übersetzt das unter src/base liegende pfc.c auf einem Unix-System (dazu benötigt man libpcap) und übersetzt den Ausdruck dann auf diesem System. Das Ergebnis (eine Menge Zeilen mit jeweils 4 Zahlen) kopiert man dann nach opt/etc/ppp/filter und baut eine neue floppy.

B.3. DYNDNS

B.3.1. Hinzufügen von neuen Providern

Das Hinzufügen von neuen Providern ist eigentlich sehr leicht, da die Update-Scripts komplett von den Provider-Daten getrennt sind. Für einen neuen Provider müssen folgende Dateien angepasst werden:

Datei opt/etc/dyndns/provider.NAME

Dies ist die Datei, in der definiert wird, wie ein Update bei diesem speziellen Provider funktioniert. Meistens besteht sie nur aus einer Liste von Variablen, da es aber ein ganz normales Shell-Script ist, können hier auch komplexere Operationen durchgeführt werden, das sollte aber selten nötig sein. In dieser Datei können folgende Variablen benutzt werden:

$ip Die IP des Interfaces, das den dynamischen Hostnamen erhalten soll.

$host Der komplette Hostname, wie ihn der Benutzer in seiner Konfiguration angegeben hat.

$subdom Alle Komponenten des Hostnamen bis zum vorletzten Punkt (**name**.provider.dom)

$domain Die letzten beiden Komponenten des Hostnamen (name.**provider.dom**)

$provider Der symbolische Name des Providers, wie ihn der Benutzer in seiner Konfgurationsdatei angegeben hat.

$user Der Benutzername für diesen Dienst.

$pass Das dazugehörige Passwort.

Diese Variablen können zur klareren Abgrenzung gegenüber anderem Text mit geschweiften Klammern geschrieben werden, aus `$ip` wird z.B. `${ip}`. Bei Verwendung von Anführungszeichen ist zu beachten, das innerhalb von einfachen Anführungszeichen die oben genannten Variablen *nicht* expandiert werden, bei doppelten Anführungszeichen schon. Als Faustregel kann man also sagen: Immer einfache Anführungszeichen benutzen, aber sobald man Variablen benutzt, doppelte Anführungszeichen benutzen.

Die folgenden Variablen müssen in dieser Datei definiert werden, damit das Paket weiß, wie ein Update bei dem entsprechenden Provider funktioniert:

provider_update_type Dies bestimmt die Art der Anfrage, die an den Server des Providers geschickt wird. Momentan werden unterstützt:

- **http** Es wird automatisiert eine bestimmte Webseite des Providers abgerufen und so der DynDNS-Eintrag aktualisiert.
- **netcat** Es wird einfach ein bestimmter Text an den Server des Providers geschickt, der das Update auslöst.
- **gnudip** Ein relativ einfaches aber sicheres Updateverfahren, welches über zwei HTTP-Anfragen ausgeführt wird.

provider_host Der Hostname des Providers, der bei einem Update kontaktiert wird.

provider_port Der Port auf dem Host des Providers, der angesprochen werden soll. Der Standard-Port für HTTP ist 80.

Je nach Update-Typ müssen weitere Variablen angegeben werden:

HTTP provider_url Hier wird die relative URL (ohne Hostname, aber mit / am Anfang zum Script des Providers abgelegt. Für Beispiele bitte die Dateien der anderen Provider ansehen.

provider_auth (optional) Benötigt der Provider eine Anmeldung per Basic Authentication, so sind hier die entsprechenden Daten anzugeben. Das Format ist `"USER:PASSWORD"`.

Netcat provider_data Dies ist der Text, der an den Server des Providers geschickt wird. Siehe z.B. `provider.DYNEISFAIR`.

GNUDip provider_script Der Pfad zum GNUDip-Script auf dem Server, dies ist meist etwas wie z.B. `'/cgi-bin/gdipupdt.cgi'`.

Datei opt/dyndns.txt

Hier müssen eine oder mehr Zeilen für den neuen Provider eingefügt werden. Meistens reicht eine Zeile in der Art:

```
dyndns_%_provider   NAME   etc/dyndns/provider.NAME
```

Wird für den Provider HTTP und Basic Authentication benutzt, so braucht man noch das base64-Tool:

```
dyndns_%_provider   NAME   files/usr/local/bin/base64
```

Sollten noch andere Tools benötigt werden, bitte mir vorher eine Mail schicken, damit ich prüfen kann, ob das für das OPT_DYNDNS geeignet ist.

Datei check/dyndns.exp

In dieser Datei muss an der langen Zeile, die mit `DYNPROVIDER =` beginnt, der Providername mit einem senkrechten Strich von den anderen abgetrennt, hinten angefügt werden.

Datei doc/<Sprache>/tex/dyndns/dyndns_main.tex

In der Dokumentation einen neuen Abschnitt eintragen. Auch hier sind die Provider alphabetisch nach dem Kurznamen, den der Benutzer in der Config-Datei eingibt, sortiert. Das prov-Makro ist am Anfang der Datei dokumentiert, genug Beispiele sollten vorhanden sein.

B.3.2. Dank

Als allererstes möchte ich dem danken, der dieses Paket ins Leben gerufen hat und lange Zeit dieses Paket betreut hat: Thomas Müller (E-Mail: `opt_dyndns@s2h.cx`) hat hier hervorragende Arbeit geleistet, ohne ihn wäre das Paket in der heutigen Form nicht möglich gewesen.

Desweiteren möchte ich auch Marcel Döring (E-Mail: `m@rcel.to`) danken, der das Paket einige Zeit lang gepflegt hat.

Bei der Entwicklung des Paketes haben sehr viele Leute geholfen und Ideen beigetragen. Mein Dank gilt allen diesen fleißigen Helfern.

Außerdem danke ich Frank Meyer und dem Rest des FLI4L-Tems für ihre unermüdliche Arbeit, um den besten Ein-Disketten-Router der Welt zu basteln (Bitte nicht ganz Ernst nehmen ;-).

Weiterhin möchte ich folgenden Leuten danken, die sich mit Tips, neuen Providern, Fehlerberichten, etc. an dem Paket beteiligt haben:

- Paul Bischof für den Provider AFRAID.

Bevor ich das Paket übernommen habe, haben schon sehr viele Leute an diesem Paket mitgewirkt. Natürlich soll ihr Beitrag nicht in Vergessenheit geraten, daher zitiere ich hier die Dankesliste, die Thomas Müller (d.h. ich = Thomas) in seinem Paket hatte:

- Jens Fischer schrieb das Paket opt_dtdns, welches mich erst auf die Idee brachte, ein Paket für DynDNS.org zu schreiben.
- Till Jäger schrieb das Paket opt_cjb, welches in in opt_dyndns übernommen habe.
- Tobias Gruetzmacher hat auf http://portfolio16.de/index.de Informationen zu weiteren DynDNS-Anbietern zusammengetragen, die ich hier verwendet habe.
- Die Anbieter dynamischer DNS, die auf ihren Webseiten zum Teil sehr gute, zum Teil weniger gute Beschreibungen des zu verwendenden Protokolls veröffentlicht haben.
- Die Programmierer diverser Update-Programme für DynDNS Anbieter, aus deren Code ich schamlos geklaut habe. ;-)
- Heiko Ambos von dynaccess.de hat mich bei der Entwicklung der Unterstützung für diesen Anbieter unterstützt.
- Dennis Neuhäuser, der die Idee hatte, die Antworten der Dienste per Webserver verfügbar zu machen statt sie auf der Konsole auszugeben (geniale Idee, wieso bin ich nicht selbst darauf gekommen?), und mir auch gleich eine Implementation dafür geschickt hat (die ich dann umgehend so weit aufgebohrt habe, dass er sie vermutlich nicht mehr wiedererkennt).
- Lars Winkler der freundlicherweise die Änderungen, um das Paket unter 2.0pre2 zum Laufen zu bringen zur Verfügung gestellt hat.
- Markus Kraft und Tobias Gruetzmacher haben die Grundlage für die Anpassung an FLI4L 2.0 gelegt.
- Diverse andere Leute haben mir ihre jeweilige Portierung auf FLI4L 2.0 geschickt. Ich muss zu meiner Schande gestehen, dass ich mir die wenigsten davon angesehen habe.
- Georg Bärwald für die Daten zu Selfhost.de
- Mark C. Storck für die Daten zu Storck.org
- Arne Biermann für den Hinweis auf den Anbieter hn.org
- Detlef Paschke für die Daten zu dyn.ee und dyndns.dk
- Martin Kisser für seine Idee zum Vermeiden von Updates, wenn die IP sich nicht geändert hat.
- Björn Hoffmann für die Daten von DnsArt.com
- Christian Busch für die Daten von no-ip.com.

- Ralf Gill für das Update der Daten von selfhost.de.
- Michael (HeinB) für eine weitere Möglichkeit sich mit fli4l selbst in den Fuss zu schiessen. ;-)
- Marcus Mönnig, dito.

B.3.3. Lizenz

Copyright ©2001-2002 Thomas Müller (E-Mail: `opt_dyndns@s2h.cx`)
Copyright ©2002-2003 Tobias Gruetzmacher (E-Mail: `fli4l@portfolio16.de`)
Copyright ©2004-201x fli4l-Team (E-Mail: `team@fli4l.de`)

Dieses Programm ist freie Software. Sie können es unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation herausgegeben, weitergeben und/oder modifizieren, entweder unter Version 2 der Lizenz oder (wenn Sie es wünschen) jeder späteren Version.

Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, dass es Ihnen von Nutzen sein wird, aber OHNE JEDE GEWÄHRLEISTUNG - sogar ohne die implizite Gewährleistung der MARKTREIFE oder der EIGNUNG FÜR EINEN BESTIMMTEN ZWECK. Details finden Sie in der GNU General Public License.

Sie sollten eine Kopie der GNU General Public License zusammen mit diesem Programm erhalten haben. Falls nicht, schreiben Sie an die

```
Free Software Foundation Inc.
59 Temple Place
Suite 330
Boston MA 02111-1307 USA.
```

Der Text der GNU General Public License ist auch im Internet unter http://www.gnu.org/licenses/gpl.txt veröffentlicht. Eine inoffizielle deutsche Übersetzung findet sich unter http://www.gnu.de/documents/gpl.de.html Diese Übersetzung soll jedoch nur zu einem besseren Verständnis der GPL verhelfen, rechtsverbindlich ist die englischsprachige Version.

B.4. EASYCRON - Crontab in der Boot-Phase ergänzen

Hinweis: *Die folgenden Ausführungen richten sich nur an Entwickler von Opt-Paketen für den fli4l Router.*

Ab Version 2.1.7 stellt das rc-Script von Easycron die Funktion `add_crontab_entry()` zur Verfügung. Durch Aufruf dieser Funktion können andere rc-Scripts ab Startposition rc101 bis Startposition 949 Einträge an die Crontab anhängen. Am Ende der Boot-Phase mit dem Starten des cron Daemon sind die zusätzlichen Einträge aktiv.

`add_crontab_entry time command [custom]`

Mit `time` wird die Ausführungszeit in cron Notation übergeben, siehe die Manpage zu crontab(5) (http://linux.die.net/man/5/crontab). `command` enthält den auszuführenden Befehl. Der dritte Parameter `custom` ist optional. Mit ihm können Umgebungsvariablen passend zum Befehl gesetzt werden. Soll mehr als eine Variable gesetzt werden, sind die Zuweisungen

durch \\ zu trennen. Bitte **nicht** die Umgebungsvariable `PATH` verändern, da sonst nachfolgende crontab Einträge nicht mehr korrekt abgarbeitet werden können.

```
#
# example I: normal use, 2 parameters
#
     crontime="0 5 1 * *"
     croncmd="rotate_i_log.sh"

     add_crontab_entry "$crontime" "$croncmd"

#
# end of example I
#

#
# example II: extended use, 3 parameters and
#                            multiple environment values
#
     croncustom="source=/var/log/current \\ dest=/mnt/data/log"
     croncmd='cp $source $dest-`date +\%Y\%m\%d`; > $source'
     crontime="59 23 * * *"

     add_crontab_entry "$crontime" "$croncmd" "$croncustom"

#
# end of example II
#
```

Die Richtigkeit der Einträge muß das aufrufende Script sicherstellen.

B.5. HD - Fehler im Zusammenhang mit Festplatten/CompactFlashs

Problem:

- der Router erkennt die Festplatte überhaupt nicht.

Mögliche Ursachen:

- das `OPT_HDDRV` wurde nicht konfiguriert und damit fehlen dem Router die Treiber für den HD-Controller
- Platte ist falsch in BIOS eingetragen.
- der IDE-Controller ist defekt oder abgeschaltet.
- es wird bei der Installation die falsche Platte angegeben (z.B. wird eine Festplatte am zweiten IDE-Kanal als hdc oder hdd angesprochen)

- der IDE-Controller wird nicht von fli4l unterstützt. Aktuelle UDMA-Controller benötigen spezielle Treiber, die in fli4l nicht enthalten sind

Problem:

- die Installation bricht ab
- nach einem Remote-Update des opt-Archives bootet der Router nicht mehr
- es gibt Fehlermeldungen beim Partitionieren oder Formatieren der Festplatte

Mögliche Ursachen:

- bei neuen Ultra-DMA-Festplatten könnte es an zu langen oder ungeeigneten IDE-Kabeln liegen
- bei älteren Festplatten ist die Einstellung der Transferrate/PIO-Mode im Bios oder auf dem Controler evtl. zu schnell für die Platte.
- ungeigneter IDE-Chipsatz

Bemerkungen:

- wenn der Router nur nicht mehr vom Netzwerk aus ansprechbar ist und man die Notfall-option zusätzlich installiert hat, kann man vor Ablauf der Wartezeit den Router mit der Reset-Tase neu starten (oder einfach aus- und einschalten). Dann wird mit der Notfall-Version gestartet und man kann ein korrigiertes opt-Archive aufspielen.

Problem:

- der Router arbeitet nach der Installation auf Festplatte nicht mehr so wie nach der Installation auf Diskette

Mögliche Ursache:

- es werden soviele Pakete benutzt, daß die Resourcen des Rechners nicht ausreichen (zu-wenig RAM, CPU, andere Hardwareprobleme)

Bemerkungen:

- gerade bei Verwendung einer Swap-Partition ist die einwandfreie Funktion der Festplatte und des Controllers sehr wichtig.

Problem:

- nach der Installation bootet fli4l nicht von Festplatte

Mögliche Ursache:

- falls nur noch die Buchstaben LI erscheinen, war auf der Festplatte vorher Linux mit LILO installiert. Es sind noch Reste vom LILO im Master-Boot-Record zu finden. Mit einer DOS-Bootdisk und dem Befehl “fdisk /mbr” kann man diese Reste entfernen.

- wenn der Bootvorgang von einem CF-Modul fehlschlägt sollte man prüfen ob das CF-Modul im Bios mit LBA oder LARGE erkannt wurde. Die richtige Einstellung für Module unter 512MB ist NORMAL oder CHS.
- es wird ein Adaptec 2940 Controller mit altem BIOS eingesetzt und das erweiterte Mapping für Festplatten über 1GB ist aktiv. Als Abhilfe kann man das BIOS des SCSI-Controllers aktualisieren oder das Mapping umschalten. **Beim Umschalten des Mappings gehen alle Daten auf der Platte verloren!**

Problem:

- am Router erscheinen Fehlermeldungen “hda: lost interrrupt”
- nach einiger Betriebsdauer treten Filesystemfehler auf bis hin zu kaputten Partitionstabellen, was eine Neuinstallation notwendig macht.

Mögliche Ursache:

- im BIOS wurde die Platte zu schnell eingestellt. In diesem Fall hilft wohl die Einstellung im BIOS zu verlangsamen.

Problem:

- Bei der Installation auf DiskOnChip werden falsche Partitionsgrößen angelegt. Der Router bootet nicht von DoC.

Mögliche Ursache:

- Nachdem ein erster Installationsversuch fehlgeschlagen ist (evtl. zu große Partitionen angegeben o.ä.) wurde die Installation ein weiteres mal gestartet ohne zwischendurch neu von der setup-Diskette zu booten. Nach einer missglückten Installation sollte man immer neu booten, damit keine störenden Reste wie z.B. noch gemountete Partitionen vom ersten Versuch den zweiten auch wieder fehlschlagen lassen.

Problem:

- Windows sagt während des Erstellens einer CF-Card: „Medium im Laufwerk (X:) bestitzt kein FAT. [Abbruch]“.

Mögliche Ursache:

- Die Compactflash wurde zu früh / ohne Abmeldung aus dem Reader entfernt. Windows hatte den letzten Schreibvorgang noch nicht abgeschlossen, das Dateisystem ist nun beschädigt. Erstelle die CompactFlash nochmals direkt am Fli4l mittels HD-Install.

B.6. HTTPD

B.6.1. Zusätzliche Einstellungen

Diese Einstellungen stehen normalerweise nicht in der Konfigurationsdatei, müssen also hinzugefügt werden, wenn sie benötigt werden.

HTTPD_USER Mit dieser Option ist es möglich, den Webserver mit den Rechten eines anderen Benutzers als „root" laufen zu lassen. Dies ist besonders sinnvoll, wenn der Webserver benutzt wird, um andere Seiten als das Admin-Interface bereitzustellen. Achtung: Es kann sein, dass einige Scripts, die Zugriff auf Konfigurationsdateien brauchen, dann nicht mehr laufen. Die Standard-Scripts dieses Pakets laufen unter jedem Benutzer.

B.6.2. Allgemeine Bemerkungen

Wenn man TELMOND installiert hat, werden auf der Status- und der Calls- Seite die Telefonnummern der Anrufer angezeigt. Eine Namenszuordnung lässt sich in der Datei opt/etc/phonebook vornehmen. Diese Datei hat das gleiche Format wie die Telefonnummerndatei vom IMONC. Es können also Telefonbücher zwischen IMONC und Router ausgetauscht werden. Das Format jeder Zeile ist dabei "Telefonnummer=Name[,WAV-Datei]" (ohne die Anführungszeichen). Die WAV-Datei wird aber nur vom IMONC benutzt und vom Webserver ignoriert.

Das komplette Webinterface ist seit der Version 2.1.12 auf ein Framefreies Design mit CSS umgestellt worden. Alte Browser könnten damit Probleme haben. Allerdings hat das den Vorteil, dass man das Aussehen der Oberfläche fast beliebig verändern kann, einfach indem man die CSS-Dateien (im wesentlichen /opt/srv/www/css/main.css) anpasst.

Das Webserver-Paket wurde von Thorsten Pohlmann (E-Mail: `pohlmann@tetronik.com`) erstellt und wird zur Zeit von Tobias Gruetzmacher (E-Mail: `fli4l@portfolio16.de`) gepflegt. Das neue Design (seit der Version 2.1.12) wurde von Helmut Hummel (E-Mail: `hh@fli4l.de`) realisiert.

B.7. IPV6 - Anbindung ans IPv6-Internet mit Hilfe eines SixXS-Tunnels

In diesem Abschnitt wird beschrieben, wie das IPv6-Paket genutzt werden kann, um mit Hilfe eines Tunnels des Anbieters SixXS (https://www.sixxs.net/) das eigene Heimnetzwerk mit dem IPv6-Internet zu verbinden.

B.7.1. Account erstellen

Zuerst muss ein SixXS-Account unter "Signup for new users" beantragt werden. Hat man diese Hürde genommen, besitzt man einen Benutzernamen der Form YYYYY-SIXXS sowie ein zugehöriges Passwort. Diese Daten benötigt man später für die Tunnelkonfiguration.

B.7.2. Tunnel konfigurieren

Vorbereitungen

Doch vorher muss man den Tunnel beantragen. Dies geschieht nach der Anmeldung über den Menüpunkt "Request tunnel". Hier ist es wichtig, beim Tunneltyp den zweiten Eintrag, "Dynamic IPv4 Endpoint using Heartbeat protocol" auszuwählen, weil diese Konfiguration vom fli4l direkt unterstützt wird. Die dritte Variante, "Static IPv4 Endpoint", ist ebenfalls

möglich, wenn man eine fest zugeordnete IPv4-Adresse sein Eigen nennt, die sich nie ändert. Die Tunnelvariante "Dynamic NAT-traversing IPv4 Endpoint using AYIYA" wird derzeit vom IPv6-Paket nicht unterstützt.

Sobald man in den anderen Feldern Angaben zum Ort des Routers gemacht und via "Next step" zur zweiten Seite gewechselt hat, stehen hier ein oder mehrere PoPs (Points of Presence) zur Auswahl, die später für den Tunnelaufbau wichtig sind. Man sollte denjenigen nehmen, der am dichtesten ist, um das Tunneln von IPv6-Paketen möglichst effizient zu machen.

Sind alle Angaben gemacht und via "Place request for new Tunnel" abgeschickt worden, kommt irgendwann darauf eine E-Mail mit den nötigen Tunneldaten an. Dazu gehören:

1. die Identifikationsnummer des Tunnels (T...)
2. der Name des zugeordneten PoPs
3. die IPv4-Adresse des zugeordneten PoPs ("SixXS IPv4")
4. die lokale IPv6-Adresse des Tunnels inklusive Subnetzmaske (bei SixXS typischerweise /64), also die Adresse des Routers ("Your IPv6")
5. die entfernte IPv6-Adresse des Tunnels inklusive Subnetzmaske (die mit der Subnetzmaske der lokalen IPv6-Adresse identisch ist), d.h. die Adresse des PoPs ("SixXS IPv6")

Konfiguration

Jetzt kann der Tunnel konfiguriert werden! Als erstes wird die Variable `IPV6_TUNNEL_N` auf "1" gesetzt, weil genau ein Tunnel aufgebaut werden soll:

```
IPV6_TUNNEL_N='1'
```

Die SixXS-Angaben werden wie folgt in der IPv6-Konfiguration vermerkt:

1. Die Identifikationsnummer des Tunnels wird in `IPV6_TUNNEL_1_SIXXS_ID` eingetragen.
2. entfällt (der Name des PoPs ist uninteressant)
3. Die IPv4-Adresse des PoPs wird in der Variable `IPV6_TUNNEL_1_REMOTEV4` vermerkt.
4. Die lokale IPv6-Adresse des Tunnels landet inklusive Subnetzmaske in der Variable `IPV6_TUNNEL_1_LOCALV6`, .
5. Die entfernte IPv6-Adresse des Tunnels wird inklusive Subnetzmaske als Teil der Default-Route (für das Zielnetz `::/0`) als Gateway in der Variable `IPV6_ROUTE_1` eingetragen (siehe Beispiel, *ohne* die spitzen Klammern um "SixXS IPv6" einzutragen):

   ```
   IPV6_ROUTE_N='1'
   IPV6_ROUTE_1='::/0 <entfernte IPv6-Adresse des Tunnels>/64'
   ```

Zusätzlich müssen Benutzername und Passwort in der Tunnelkonfiguration in den Variablen `IPV6_TUNNEL_1_SIXXS_USERNAME` und `IPV6_TUNNEL_1_SIXXS_PASSWORD` angegeben werden. Schließlich sollte in der Variablen `IPV6_TUNNEL_1_TYPE` vermerkt werden, dass der konfigurierte Tunnel ein SixXS-Heartbeat-Tunnel ist:

```
IPV6_TUNNEL_1_TYPE='sixxs-heartbeat'
```

Hat man von SixXS den PoP "deham01" mit der IPv4-Adresse `212.224.0.188` sowie die Tunnelendpunkte `2001:db8:900:551::1/64` (entfernt) und `2001:db8:900:551::2/64` (lokal) erhalten, und lautet die Tunnel-ID "T1234", der Benutzername "USER1-SIXXS" und das Passwort "sixxs" (bitte dieses Passwort *nicht* verwenden!), dann sieht die resultierende Konfiguration so aus:

```
IPV6_TUNNEL_N='1'
IPV6_TUNNEL_1_LOCALV4='dynamic' # oder feste lokale IPv4-Adresse
IPV6_TUNNEL_1_REMOTEV4='212.224.0.188'
IPV6_TUNNEL_1_LOCALV6='2001:db8:900:551::2/64'
IPV6_TUNNEL_1_TYPE='sixxs-heartbeat'
IPV6_TUNNEL_1_SIXXS_USERNAME='USER1-SIXXS'
IPV6_TUNNEL_1_SIXXS_PASSWORD='sixxs'
IPV6_TUNNEL_1_SIXXS_ID='T1234'

IPV6_ROUTE_N='1'
IPV6_ROUTE_1='::/0 2001:db8:900:551::1/64'
```

Test

Hat man dies geschaffft, dann kann man nach Aktualisierung der Konfiguration den fli4l-Router neu starten. Nach dem Einloggen auf dem Router (direkt oder z.B. per SSH) sollte man den Tunnelendpunkt bereits anpingen können. Das Ganze sieht dann mit den oben genannten Beispiel-Daten folgendermaßen aus:

```
garm 3.6.0-revXXXXX # ping -c 4 2001:db8:900:551:0:0:0:1
PING 2001:db8:900:551::1 (2001:db8:900:551::1): 56 data bytes
64 bytes from 2001:db8:900:551::1: seq=0 ttl=64 time=67.646 ms
64 bytes from 2001:db8:900:551::1: seq=1 ttl=64 time=72.001 ms
64 bytes from 2001:db8:900:551::1: seq=2 ttl=64 time=70.082 ms
64 bytes from 2001:db8:900:551::1: seq=3 ttl=64 time=67.996 ms

--- 2001:db8:900:551::1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 67.646/69.431/72.001 ms
```

Wichtig ist in der vorletzten Zeile der Teil "0% packet loss", d.h. dass für alle PING-Pakete Antwortpakete empfangen wurden. Kommt keine Antwort vom anderen Ende des Tunnels, sieht das Ergebnis anders aus:

```
garm 3.6.0-revXXXXX # ping -c 4 2001:db8:900:551:0:0:0:1
PING 2001:db8:900:551::1 (2001:db8:900:551::1): 56 data bytes

--- 2001:db8:900:551::1 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
```

Hier fällt auf, dass für kein einziges PING-Paket eine Antwort empfangen wurde ("100% packet loss"). Das bedeutet, dass entweder die Konfiguration nicht korrekt ist, oder dass der Tunnel seitens SixXS noch nicht vollständig eingerichtet worden ist. Im zweiten Fall sollte

man erst einige Zeit abwarten, weil die Konfiguration auf Seiten des PoPs durchaus ein paar Stunden dauern kann. Hat man die Konfiguration doppelt und dreifach geprüft und keinen Fehler entdeckt, und ist einige Zeit verstrichen, ohne dass der Tunnel funktioniert, sollte man sich per E-Mail an SixXS wenden und das Problem möglichst detailliert beschreiben.

B.7.3. Subnetz konfigurieren

Vorbereitungen

Funktioniert der Tunnel, hat man den ersten großen Punkt geschafft. Fertig ist man aber noch nicht. Denn nun hat der Router zwar die Möglichkeit, Pakete ins IPv6-Internet zu schicken und von dort zu empfangen, aber die Hosts im lokalen Netz noch nicht. Dafür muss erst ein IPv6-Subnetz konfiguriert werden, innerhalb dessen die Hosts eingebunden werden.

Hier fällt ein kleiner, aber wesentlicher Unterschied zur Konfiguration eines IPv4-Netzwerks auf: Wegen der Adressknappheit ist in der Regel nur ein Host direkt mit dem Internet verbunden. Die anderen Hosts im lokalen Netz erhalten nur netzinterne, d.h. nicht nach außen geroutete Adressen aus den Bereichen `192.168.*.*`, `172.16.*.*` bis `172.31.*.*` sowie `10.*.*.*`, je nach Größe des Subnetzes. Bei IPv6 gibt es Adressen in Hülle und Fülle, somit entfällt die Notwendigkeit, netzinterne Adressen zu verwenden. Wegen des globalen Charakters lokaler Subnetze muss jedoch sichergestellt werden, dass die Adressen lokaler Hosts nicht mit anderen Adressen im Internet kollidieren. Deshalb muss ein Subnetz vom IPv6-Anbieter zugeteilt werden, um solche Kollisionen zu vermeiden.

Bei SixXS geschieht dies mit dem Menüpunkt “Request subnet”. Hier muss man hauptsächlich den zu verwendenden Tunnel angeben, was leicht ist, weil bisher nur einer konfiguriert worden ist. Nach dem Abschicken des Formulars via “Place request for new subnet” erhält man nach einiger Zeit eine E-Mail mit den folgenden Informationen:

1. Die IPv6-Adresse des Subnetzes inklusive Subnetzmaske (“Subnet IPv6”)
2. Die IPv6-Adresse des Routers im Tunnel, wohin das Subnetz seitens SixXS geroutet wird (“Routed to”)
3. Die IPv4-Adresse des Routers (“Your IPv4”)[1]

Diese Daten reichen aus, beim fli4l jetzt ein eigenes IPv6-Subnetz zu konfigurieren. Allerdings muss man eines noch wissen: Das zugewiesene Subnetz ist in der Regel sehr groß. SixXS teilt für gewöhnlich /48er-Subnetze zu, d.h. innerhalb der 128 Bit langen IPv6-Adresse belegt der Anteil, der auf das Netzpräfix fällt, 48 Bit, und der Anteil, der für die Adressierung der Hosts zur Verfügung steht, beträgt 128 - 48 = 80 Bit. Ein solch großes Subnetz hat jedoch zwei große Nachteile. Der erste Nachteil ist die schiere Größe: Man kann in dem Netz $2^{80} \approx 1209$ Trilliarden Hosts unterkriegen. Das zu verwalten, ohne eine weitere Struktur auf dem Hosts-Anteil der Adresse zu nutzen, erscheint nicht ratsam. Der zweite Nachteil wiegt schwerer: Innerhalb eines solch großen Subnetzes funktioniert die so genannte *IPv6-Autokonfiguration* nicht mehr. Das ist ein Vorgang, bei dem ein IPv6-Host über bestimmte Protokolle das Subnetz-Präfix erhält und sich seine IPv6-Adresse mit Hilfe der MAC-Adresse seines Netzwerk-Adapters zurechtbastelt. Die MAC-Adresse besteht aus sechs Bytes, und mit Hilfe des Standards EUI-64 kann man sie auf acht Bytes strecken. Das entspricht 64 Bit, und dann ist Schluss. Für 80 Bit stehen einfach nicht genügend Informationen seitens des Hosts zur Verfügung.

[1] falls der Router die IPv4-Adresse dynamisch erhält, steht hier “heartbeat”

Lange Rede, kurzer Sinn: Das Subnetz muss kleiner gemacht werden, und zwar muss, damit später die Autokonfiguration auch funktionieren kann, daraus ein /64er-Subnetz werden. Das ist ganz einfach: Die Subnetzmaske wird einfach zu /64 abgeändert. Wurde also von SixXS z.B. das Subnetz `2001:db8:123::/48` zugewiesen, dann ist das Subnetz, für das der fli4l konfiguriert werden soll, einfach `2001:db8:123::/64`. Das bedeutet im Detail, dass das /48-Subnetz in $2^{(64-48)} = 2^{16} = 65536$ Sub-Subnetze eingeteilt wird, von denen das erste mit der Nummer Null vom fli4l verwendet werden soll. Dazu muss man sich erinnern, dass die Kurzform `2001:db8:123::` eigentlich für die Adresse `2001:db8:123:0:0:0:0:0` steht, wobei die ersten drei Zahlen die vom IPv6-Anbieter global eindeutig vergebenen Teile des Subnetzes sind, die vierte Zahl das eigens ausgesuchte Sub-Subnetz "Null" darstellt, und die letzten vier Zahlen für den Host-Anteil reserviert sind. Das ergibt dann zwar immer noch ein riesiges (Sub-)Subnetz, in dem bis zu $2^{64} \approx 18,4$ Trillionen Hosts untergebracht werden können. Dank der IPv6-Autokonfiguration kommt man aber mit den tatsächlichen Adressen nicht in Berührung. Und das ist gut so...

Konfiguration

Zurück zur Konfiguration! Als erstes wird die Variable `IPV6_NET_N` auf "1" gesetzt, weil genau ein lokales IPv6-Subnetz eingerichtet werden soll. Die IPv6-Adresse des /64-Subnetzes landet inklusive Subnetzmaske in der Variable `IPV6_NET_1`. Doch das ist nicht ganz richtig: Vielmehr landet die IPv6-Adresse *des Routers innerhalb dieses Subnetzes* in der Variable `IPV6_NET_1`. Somit dient die Variable zwei Zwecken: Zum einen wird das IPv6-Subnetz definiert, und zum anderen wird gleichzeitig festgelegt, welche Adresse in diesem Subnetz dem Router selbst zugeordnet ist. Damit schlägt man gleich zwei Fliegen mit einer Klappe.

Hat man nun das /48er-IPv6-Subnetz `2001:db8:123::/48` von SixXS erhalten, daraus das erste (Index Null) als zu verwendendes /64er-Sub-Subnetz ausgewählt und schließlich bestimmt, dass der Router innerhalb dieses Subnetzes die Adresse "1" erhalten soll, dann lautet die Adresse des Routers somit `2001:db8:123::1`. Wir erhalten somit die folgende Konfiguration:

```
IPV6_NET_N='1'
IPV6_NET_1='2001:db8:123::1/64' # IPv6-Adresse des Routers + Subnetzmaske
```

Jetzt fehlt noch der Name der Netzwerkschnittstelle, an die dieses Subnetz gebunden werden soll. Jedes Subnetz wird genau einer Netzwerkschnittstelle zugeordnet. Falls sich nur eine konfigurierte Netzwerkkarte im Router befindet, so lautet der Name der Netzwerkschnittstelle typischerweise "eth0" für kabelgebundene oder "wlan0" für drahtloses Adapter. Schauen Sie im Zweifelsfall einfach die Einstellung für `IP_NET_1_DEV` ("IP" ohne "6") und kopieren Sie den Inhalt einfach herüber:

```
IPV6_NET_1_DEV='eth0' # Netzwerkschnittstelle für dieses IPv6-Subnetz
```

Schließlich müssen wir nur noch alle Hebel der IPv6-Autokonfiguration ziehen:

```
IPV6_NET_1_ADVERTISE='yes' # Subnetz-Präfix und Default-Route
IPV6_NET_1_ADVERTISE_DNS='yes' # DNS-Server (nur wenn DNS_SUPPORT_IPV6='yes'!)
IPV6_NET_1_DHCP='yes' # Domänen-Name und DNS-Server
```

Die beiden letzten Einstellungen sind nicht zwingend notwendig für ein funktionierendes IPv6-Subnetz, sind aber ganz hilfreich. Sie dienen der Verbreitung zusätzlicher Informationen im IPv6-Subnetz, nämlich der IPv6-Adresse des DNS-Servers sowie der verwendeten Domäne. Den DNS-Server kann man sogar auf zweierlei Weise veröffentlichen. Weil verschiedene Systeme hier unterschiedliche Vorlieben an den Tag legen, ist es von Vorteil, beide Verfahren (RDNSS via Router Advertisments sowie DHCPv6) zu aktivieren.

Test

Die gesamte IPv6-Konfiguration dieses Beispiels lautet:

```
IPV6_NET_N='1'
IPV6_NET_1='2001:db8:123::1/64' # IPv6-Adresse des Routers + Subnetzmaske
IPV6_NET_1_DEV='eth0' # Netzwerkschnittstelle für dieses IPv6-Subnetz
IPV6_NET_1_ADVERTISE='yes' # Subnetz-Präfix und Default-Route
IPV6_NET_1_ADVERTISE_DNS='yes' # DNS-Server (nur wenn DNS_SUPPORT_IPV6='yes'!)
IPV6_NET_1_DHCP='yes' # Domänen-Name und DNS-Server

IPV6_TUNNEL_N='1'
IPV6_TUNNEL_1_LOCALV4='dynamic' # oder feste lokale IPv4-Adresse
IPV6_TUNNEL_1_REMOTEV4='212.224.0.188'
IPV6_TUNNEL_1_LOCALV6='2001:db8:900:551::2/64'
IPV6_TUNNEL_1_TYPE='sixxs-heartbeat'
IPV6_TUNNEL_1_SIXXS_USERNAME='USER1-SIXXS'
IPV6_TUNNEL_1_SIXXS_PASSWORD='sixxs'
IPV6_TUNNEL_1_SIXXS_ID='T1234'

IPV6_ROUTE_N='1'
IPV6_ROUTE_1='::/0 2001:db8:900:551::1/64'
```

Ein normal konfigurierter Windows 7-Host sollte mit einem solch konfigurierten fli4l-Router automatisch seine IPv6-Adresse sowie Default-Route, DNS-Server und Domäne konfigurieren und somit den Rechner IPv6-tauglich machen. Das kann man z.B. mit einem enfachen PING vom Windows-Rechner ins IPv6-Internet realisieren. Im folgenden Beispiel versuchen wir, vom Windows-Host aus den fli4l.de-Webserver zu erreichen (wir benutzen dabei direkt die IPv6-Adresse, um nicht von der Korrektheit der DNS-Funktionalität auszugehen):

```
C:\>ping 2001:bf0:c000:a::2:132

Ping wird ausgeführt für 2001:bf0:c000:a::2:132 mit 32 Bytes Daten:

Antwort von 2001:bf0:c000:a::2:132: Zeit=104ms
Antwort von 2001:bf0:c000:a::2:132: Zeit=102ms
Antwort von 2001:bf0:c000:a::2:132: Zeit=106ms
Antwort von 2001:bf0:c000:a::2:132: Zeit=106ms

Ping-Statistik für 2001:bf0:c000:a::2:132:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 102ms, Maximum = 106ms, Mittelwert = 104ms
```

Schließlich kann man das Werkzeug “traceroute” (unter Windows: “tracert”) verwenden, um zu untersuchen, ob ein Paket korrekt geroutet wird. Ein Beispiel aus dem lokalen Netz des Autors ist untenstehend abgebildet. Daran kann man gut erkennen, dass ein Paket erst zum fli4l-Router (erste Zeile), dann zum anderen Ende des Tunnels (zweite Zeile) und schließlich in das weltweite IPv6-Internet (ab der dritten Zeile) gelangt:

```
C:\>tracert 2001:bf0:c000:a::2:132

Routenverfolgung zu virtualhost.in-berlin.de [2001:bf0:c000:a::2:132]
  über maximal 30 Abschnitte:

  1    <1 ms    <1 ms    <1 ms  garm.schulz.life24h.net [2001:6f8:13da:1::1]
  2    70 ms    79 ms    71 ms  gw-1362.ham-01.de.sixxs.net [2001:6f8:900:551::1]
  3    67 ms    71 ms    76 ms  2001:6f8:800:1003::209:55
  4    68 ms     *       70 ms  2001:6f8:1:0:87:86:71:240
  5    69 ms     *       71 ms  2001:6f8:1:0:87:86:77:67
  6    72 ms     *       71 ms  2001:6f8:1:0:86:87:77:81
  7    71 ms     *       71 ms  2001:6f8:1:0:87:86:77:83
  8    90 ms     *       81 ms  2001:6f8:1:0:87:86:77:62
  9    84 ms     *       88 ms  2001:6f8:1:0:87:86:77:71
 10    99 ms    83 ms    83 ms  2001:6f8:1:0:87:86:77:249
 11    94 ms    87 ms    87 ms  20gigabitethernet4-3.core1.fra1.he.net
  [2001:7f8::1b1b:0:1]
 12    96 ms    99 ms    99 ms  10gigabitethernet1-4.core1.ams1.he.net
  [2001:470:0:47::1]
 13   105 ms   108 ms   107 ms  2001:7f8:8:5:0:73e6:0:1
 14   106 ms   107 ms   104 ms  virtualhost.in-berlin.de [2001:bf0:c000:a::2:132]

Ablaufverfolgung beendet.
```

B.8. ISDN

B.8.1. Technische Details zu Einwahl und Routing bei ISDN

Dieses Kapitel ist nur für Leute interessant, die ein wenig verstehen wollen, was intern passiert, die spezielle Konfigurationswünsche haben oder die nach der Lösung für Problem suchen. Andere sollten dieses Kapitel bitte *nicht* lesen.

Nach dem Herstellen einer Verbindung zum Provider konfiguriertder ipppd-Daemon, der diese Verbindung hergestellt hat, das Interface neu, um die ausgehandelten IP-Adressen zu setzen. Dabei werden vom Linux-Kern automatisch auch Routen gesetzt, die der Remote-IP und der Netzmaske entsprechen und vorhandene, spezielle Routen werden gelöscht. Wird keine Netzmaske vorgegeben, leitet der ipppd aus der Remote-IP die Netzmaske ab (er benutzt dazu die Überholte Einteilung in Class A,B und C Subnetze). Das Verschwinden der Routen und die automatisch gesetzten neuen Routen haben immer wieder für Probleme gesorgt:

- Firmennetze waren nicht mehr erreichbar, weil die Routen verschwunden waren oder von den gesetzten neuen Routen überlagert wurden
- Interfaces wählten sich scheinbar ohne Grund ein, da ein Paket statt über die default Route über die vom Kern generierte Route auf ein anderes Interface ging

- ...

Daher wird nunmehr versucht, diese unerwünschten Routen zu verhindern.
Dazu werden folgende Dinge geändert:

- remote ip wird auf 0.0.0.0 gesetzt, wenn nichts anderes spezifiert ist. Dadurch verschwinden die Routen, die beim Konfigurieren des Interfaces vom Kern eingerichtet werden.
- zusätzlich angegebene Routen über den Circuit werden in einer Datei zwischengespeichert
- wird eine Netzwerkmaske für den Circuit angegeben, wird diese an den ipppd weitergereicht, damit der sie nach Aushandeln der IP für die Konfiguration des Interfaces (und damit für die Generierung von Routen) nutzt.
- nach dem Einwählen werden die zwischengespeicherten Routen des Circuits ausgelesen und wieder gesetzt (sie wurden vom Kern beim Neukonfigurieren des Interfaces durch ipppd gelöscht)
- nach dem Auflegen wird das Interface wieder neu konfiguriert und die Routen werden neu gesetzt um die Ausgangssituation wieder herzustellen.

Die Konfiguration der Circuits sieht dann wie folgt aus:

- item default route

  ```
  ISDN_CIRC_%_ROUTE='0.0.0.0'
  ```

  Ist der Circuit ein lcr circuit und gerade "aktiv", wird eine default route auf diesen Circuit (bzw. das dazugehörige Interface) gesetzt. Nach dem Einwählen erscheint eine Host-Route zum Provider, die nach dem Auflegen wieder verschwindet.

- spezielle Routen

  ```
  ISDN_CIRC_%_ROUTE='network/netmaskbits'
  ```

  Es werden die angegeben Routen auf den Circuit (bzw. das dazugehörige Interface) eingerichtet. Nach dem Einwählen werden die von Kern gelöschten Routen wieder hergestellt und es gibt eine Host-Route zum Einwahlknoten. Nach dem Auflegen wird der Originalzustand wieder hergestellt.

- remote ip

  ```
  ISDN_CIRC_%_REMOTE='ip address/netmaskbits'
  ISDN_CIRC_%_ROUTE='network/netmaskbits'
  ```

  Beim Konfigurieren des Interfaces erscheinen Routen in das Zielnetz (entsprechend ipadress AND netmask). Wird die spezifizierte IP nach dem Einwählen beibehalten (daß heißt, es wird keine andere ip während des Verbindungsaufbaus ausgehandelt), bleibt die Route bestehen.

  Wird allerdings beim Einwählen eine andere IP ausgehandelt, ändert sich die Route entsprechend (new ip AND netmask).

  Für die zusätzlichen Routen gilt das oben gesagte.

Das löst hoffentlich vorläufig *alle* Probleme, die mit speziellen Routen auftraten. Die Form der Korrektur mag sich in Zukunft noch ändern, an dem Prinzip ändert sich hoffentlich nichts mehr.

B.8.2. Fehlermeldungen des ISDN-Subsystems (englisch, i4l-Dokumentation)

Im folgenden ein Auszug aus der Isdn4Linux Dokumentation (man 7 isdn_cause).

Cause messages are 2-byte information elements, describing the state transitions of an ISDN line. Each cause message describes its origination (location) in one byte, while the cause code is described in the other byte. Internally, when EDSS1 is used, the first byte contains the location while the second byte contains the cause code. When using 1TR6, the first byte contains the cause code while the location is coded in the second byte. In the Linux ISDN subsystem, the cause messages visible to the user are unified to avoid confusion. All user visible cause messages are displayed as hexadecimal strings. These strings always have the location coded in the first byte, regardless if using 1TR6 or EDSS1. When using EDSS1, these strings are preceeded by the character 'E'.

LOCATION The following location codes are defined when using EDSS1:

00 Message generated by user.
01 Message generated by private network serving the local user.
02 Message generated by public network serving the local user.
03 Message generated by transit network.
04 Message generated by public network serving the remote user.
05 Message generated by private network serving the remote user.
07 Message generated by international network.
0A Message generated by network beyond inter-working point.

CAUSE The following cause codes are defined when using EDSS1:

01 Unallocated (unassigned) number.
02 No route to specified transit network.
03 No route to destination.
06 Channel unacceptable.
07 Call awarded and being delivered in an established channel.
10 Normal call clearing.
11 User busy.
12 No user responding.
13 No answer from user (user alerted).
15 Call rejected.
16 Number changed.
1A Non-selected user clearing.
1B Destination out of order.
1C Invalid number format.
1D Facility rejected.
1E Response to status enquiry.
1F Normal, unspecified.
22 No circuit or channel available.
26 Network out of order.
29 Temporary failure.
2A Switching equipment congestion.
2B Access information discarded.
2C Requested circuit or channel not available.
2F Resources unavailable, unspecified.
31 Quality of service unavailable.
32 Requested facility not subscribed.

39 Bearer capability not authorised.
3A Bearer capability not presently available.
3F Service or option not available, unspecified.
41 Bearer capability not implemented.
42 Channel type not implemented.
45 Requested facility not implemented.
46 Only restricted digital information bearer.
4F Service or option not implemented, unspecified.
51 Invalid call reference value.
52 Identified channel does not exist.
53 A suspended call exists, but this call identity does not.
54 Call identity in use.
55 No call suspended.
56 Call having the requested call identity.
58 Incompatible destination.
5B Invalid transit network selection.
5F Invalid message, unspecified.
60 Mandatory information element is missing.
61 Message type non-existent or not implemented.
62 Message not compatible with call state or message or message type non existent or not implemented.
63 Information element non-existent or not implemented.
64 Invalid information element content.
65 Message not compatible.
66 Recovery on timer expiry.
6F Protocol error, unspecified.
7F Inter working, unspecified.

B.9. UMTS

B.9.1. Unterstützte Hardware

```
Dieses Paket unterstützt folgende UMTS-Hardware:
Für den Betrieb sind unter anderem auch weitere Pakete erforderlich.
Im USB-Paket sollten sicherheitshalber alle Lowlevel-Treiber geladen werden.
USB_LOWLEVEL='ehci uhci ohci'

Hardware:               getestet        zusätzliche Pakete

Novatel Adapter:
Merlin U530             ja              PCMCIA, TOOLS (serial)
Merlin U630             nein            PCMCIA, TOOLS (serial)

MC950D                  ja              USB


OPTION Adapter:
3G Datacard             nein            PCMCIA, USB
GT 3G Quad              ja              PCMCIA, USB
GT Fusion               nein            PCMCIA, USB
```

```
GT MAX HSUPA GX0301     ja              PCMCIA, USB
bei den vier Cardbusadaptern ist PCMCIA_PCIC='yenta_socket' zu setzen

Icon 225 (GIO225)       ja              USB

Huawei Adapter:
E220, E230, E270        ja              USB
E510                    ja              USB
E800                    ja              USB
K3520                   ja              USB

ZTE Adapter:
MF110                   ja              USB
MF190                   ja              USB
```

B.9.2. Modemschnittstelle nicht aktiviert

Bei einigen OPTION UMTS Sticks kann es vorkommen, das die Modemschnittstelle nicht aktiviert ist, welche aber für den pppd benötigt wird.

```
Beispiel anhand des GIO225 Adapter

Kontrolle mittels:
grep "" /sys/bus/usb/devices/*/tty/*/hsotype

Die Ausgabe sollte etwa so aussehen:
/sys/bus/usb/devices/2-1:1.0/tty/ttyHS0/hsotype:Control
/sys/bus/usb/devices/2-1:1.0/tty/ttyHS1/hsotype:Application
/sys/bus/usb/devices/2-1:1.1/tty/ttyHS2/hsotype:Diagnostic

Hier fehlt die Ausgabe hsotype:Modem.

Jetzt kann man mit folgenden Befehl die Interface Konfiguration kontrollieren:
chat -e -t 1 '' "AT_OIFC?" OK >/dev/ttyHS0 </dev/ttyHS0

Die Ausgabe sollte folgendermassen aussehen:
AT_OIFC?
_OIFC: 3,1,1,0

OK

Sollte dort folgendes stehen:
AT_OIFC?
_OIFC: 2,1,1,0

OK
```

```
kann man die Modemschnittstelle mit folgenden Befehl aktivieren:
chat -e -t 1 '' "AT_OIFC=3,1,1,0" OK >/dev/ttyHS0 </dev/ttyHS0

Danach den Adapter noch einmal abziehen und neu anstecken.
Jetzt sollte mittels:
grep "" /sys/bus/usb/devices/*/tty/*/hsotype

auch ein Modemeintrag vorhanden sein.
/sys/bus/usb/devices/2-1:1.0/tty/ttyHS0/hsotype:Control
/sys/bus/usb/devices/2-1:1.0/tty/ttyHS1/hsotype:Application
/sys/bus/usb/devices/2-1:1.1/tty/ttyHS2/hsotype:Diagnostic
/sys/bus/usb/devices/2-1:1.2/tty/ttyHS3/hsotype:Modem
```

C. Versionsunterschiede

C.1. Unterschiede Version 3.6.2 und 3.6.1

Package UMTS

Neue Variablen	Gelöschte Variablen
OPT_SMS (Seite ??)	
SMS_FOLDER (Seite ??)	
SMS_FOLDER_FAIL (Seite ??)	
SMS_FOLDER_SENT (Seite ??)	
SMS_LOGFILE (Seite ??)	
SMS_LOGLEVEL (Seite ??)	
SMS_RECEIVER (Seite ??)	

C.2. Unterschiede Version 3.6.2 und 3.6.0

Package UMTS

Neue Variablen	Gelöschte Variablen
OPT_SMS (Seite ??)	
SMS_FOLDER (Seite ??)	
SMS_FOLDER_FAIL (Seite ??)	
SMS_FOLDER_SENT (Seite ??)	
SMS_LOGFILE (Seite ??)	
SMS_LOGLEVEL (Seite ??)	
SMS_RECEIVER (Seite ??)	

Package USB

Neue Variablen	Gelöschte Variablen
USB_MODEM_WAITSECONDS (Seite 248)	USBMODEM_WAITSECONDS

C.3. Unterschiede Version 3.6.2 und 3.4.0

Package BASE

Umbenannte Variablen

Originaler Name	Neuer Name
POWEROFF_ON_HALT	POWERMANAGEMENT (Seite 28)
REAL_MODE_POWEROFF	POWERMANAGEMENT (Seite 28)
SER_BEEP	BEEP (Seite 30)

Neue Variablen

DEBUG_ENABLE_CORE (Seite 328)
DEBUG_IP (Seite 328)
DEBUG_IPUP (Seite 328)
DEBUG_KEEP_BOOTLOGD (Seite 328)
DEBUG_MDEV (Seite 328)
PORTFW_x_COMMENT (Seite 74)

Gelöschte Variablen

IGNORE_KSOFTIRQ

Package DHCP_CLIENT

Neue Variablen

DHCP_CLIENT_N (Seite 100)
DHCP_CLIENT_x_HOSTNAME (Seite 101)
DHCP_CLIENT_x_IF (Seite 101)
DHCP_CLIENT_x_ROUTE (Seite 101)
DHCP_CLIENT_x_USEPEERDNS (Seite 101)

Gelöschte Variablen

DHCP_CLIENT_DEV
DHCP_CLIENT_HOSTNAME
DHCP_CLIENT_IGNORE_DEF_GW
DHCP_CLIENT_INTERFACES
DHCP_CLIENT_USEPEERDNS

Package DNS_DHCP

Neue Variablen

DHCP_RANGE_x_OPTION_N (Seite 107)
DHCP_RANGE_x_OPTION_x (Seite 107)
DNS_REBINDOK_N (Seite 105)
DNS_REBINDOK_x_DOMAIN (Seite 105)

Gelöschte Variablen

Package DYNDNS

Neue Variablen

DYNDNS_x_EXT_IP (Seite 123)

Gelöschte Variablen

DYNDNS_NO_FRAMES

Package HD

Neue Variablen

EXTMOUNT_N (Seite 128)
EXTMOUNT_x_FILESYSTEM (Seite 129)
EXTMOUNT_x_MOUNTPOINT (Seite 129)
EXTMOUNT_x_OPTIONS (Seite 129)
EXTMOUNT_x_VOLUMEID (Seite 128)

Gelöschte Variablen

OPT_EXTMOUNT (Seite 128)

Package HTTPD

Neue Variablen

HTTPD_PORTFW (Seite 132)
OAC_ALL_INVISIBLE (Seite 134)
OAC_BLOCK_UNKNOWN_IF (Seite 135)
OAC_GROUP_N (Seite 135)
OAC_GROUP_x_BOOTBLOCK (Seite 135)
OAC_GROUP_x_CLIENT_N (Seite 135)
OAC_GROUP_x_CLIENT_x (Seite 135)
OAC_GROUP_x_INVISIBLE (Seite 135)
OAC_GROUP_x_NAME (Seite 135)
OAC_INPUT (Seite 134)
OAC_LIMITS (Seite 134)
OAC_MODE (Seite 135)
OAC_WANDEVICE (Seite 134)
OPT_OAC (Seite 134)

Gelöschte Variablen

Package IPV6

Neue Variablen

HOSTNAME_IP6 (Seite 137)
IPV6_MULTICAST (Seite 141)
IPV6_NET_N (Seite 137)
IPV6_NET_x (Seite 137)
IPV6_NET_x_ADVERTISE (Seite 137)
IPV6_NET_x_ADVERTISE_DNS (Seite 138)
IPV6_NET_x_DEV (Seite 137)
IPV6_NET_x_DHCP (Seite 138)
IPV6_NET_x_NAME (Seite 138)
IPV6_ROUTE_N (Seite 140)
IPV6_ROUTE_x (Seite 141)
IPV6_TUNNEL_N (Seite 139)
IPV6_TUNNEL_x_DEV (Seite 139)
IPV6_TUNNEL_x_LOCALV4 (Seite 139)
IPV6_TUNNEL_x_LOCALV6 (Seite 139)
IPV6_TUNNEL_x_MTU (Seite 140)
IPV6_TUNNEL_x_REMOTEV4 (Seite 139)
IPV6_TUNNEL_x_SIXXS_ID (Seite 140)
IPV6_TUNNEL_x_SIXXS_PASSWORD (Seite 140)
IPV6_TUNNEL_x_SIXXS_USERNAME (Seite 140)
IPV6_TUNNEL_x_TYPE (Seite 140)
OPT_IPV6 (Seite 137)
PF6_FORWARD_ACCEPT_DEF (Seite 143)
PF6_FORWARD_LOG (Seite 143)
PF6_FORWARD_LOG_LIMIT (Seite 143)
PF6_FORWARD_N (Seite 143)
PF6_FORWARD_POLICY (Seite 143)
PF6_FORWARD_REJ_LIMIT (Seite 143)
PF6_FORWARD_UDP_REJ_LIMIT (Seite 143)
PF6_FORWARD_x (Seite 144)
PF6_FORWARD_x_COMMENT (Seite 144)
PF6_INPUT_ACCEPT_DEF (Seite 141)
PF6_INPUT_LOG (Seite 142)
PF6_INPUT_LOG_LIMIT (Seite 142)
PF6_INPUT_N (Seite 142)

Gelöschte Variablen

PF6_INPUT_POLICY (Seite 141)
PF6_INPUT_REJ_LIMIT (Seite 142)
PF6_INPUT_UDP_REJ_LIMIT (Seite 142)
PF6_INPUT_x (Seite 142)
PF6_INPUT_x_COMMENT (Seite 143)
PF6_USR_CHAIN_N (Seite 144)
PF6_USR_CHAIN_x_NAME (Seite 144)
PF6_USR_CHAIN_x_RULE_N (Seite 144)
PF6_USR_CHAIN_x_RULE_x (Seite 144)
PF6_USR_CHAIN_x_RULE_x_COMMENT (Seite 144)

Package LCD

Neue Variablen

LCD_LANIP (Seite 163)
LCD_LANPASS (Seite 163)
LCD_LANTYPE (Seite 163)
LCD_LANUSER (Seite 163)

Gelöschte Variablen

Package OPENVPN

Umbenannte Variablen

Originaler Name	Neuer Name
OPENVPN_DEFAULT_MANAGMENT_LOG_CACHE	OPENVPN_DEFAULT_MANAGEMENT_LOG_CACHE (Seite 197)
OPENVPN_x_MANAGMENT_LOG_CACHE	OPENVPN_x_MANAGEMENT_LOG_CACHE (Seite 200)
OPENVPN_x_MANAGMENT_PORT	OPENVPN_x_MANAGEMENT_PORT (Seite ??)

Neue Variablen

OPENVPN_DEFAULT_PERSIST_REMOTE_IP (Seite ??)
OPENVPN_VERSION (Seite ??)

Gelöschte Variablen

OPENVPN_FEATURES

Package TOOLS

Neue Variablen

OPT_IPERF (Seite 241)
OPT_SHRED (Seite 243)
OPT_VALGRIND (Seite 244)
OPT_YTREE (Seite 244)

Gelöschte Variablen

OPT_LESS
OPT_TAIL
OPT_TOP

Package UMTS

Neue Variablen

OPT_SMS (Seite ??)
OPT_UMTS (Seite 244)

Gelöschte Variablen

SMS_FOLDER (Seite **??**)
SMS_FOLDER_FAIL (Seite **??**)
SMS_FOLDER_SENT (Seite **??**)
SMS_LOGFILE (Seite **??**)
SMS_LOGLEVEL (Seite **??**)
SMS_RECEIVER (Seite **??**)
UMTS_ADAPTER (Seite 246)
UMTS_APN (Seite 245)
UMTS_CHARGEINT (Seite 245)
UMTS_CTRL (Seite 247)
UMTS_DEBUG (Seite 244)
UMTS_DEV (Seite 247)
UMTS_DIALOUT (Seite 244)
UMTS_DRV (Seite 246)
UMTS_FILTER (Seite 246)
UMTS_GPRS_UMTS (Seite 245)
UMTS_HUP_TIMEOUT (Seite 245)
UMTS_IDDEVICE (Seite 246)
UMTS_IDDEVICE2 (Seite 246)
UMTS_IDVENDOR (Seite 246)
UMTS_IDVENDOR2 (Seite 246)
UMTS_NAME (Seite 245)
UMTS_PASSWD (Seite 245)
UMTS_PIN (Seite 244)
UMTS_SWITCH (Seite 246)
UMTS_TIMES (Seite 245)
UMTS_USEPEERDNS (Seite 245)
UMTS_USER (Seite 245)

Package USB

Neue Variablen

USB_MODEM_WAITSECONDS (Seite 248)

Gelöschte Variablen

USBMODEM_WAITSECONDS

Package WLAN

Neue Variablen

WLAN_REGDOMAIN (Seite 250)
WLAN_x_DIVERSITY (Seite 254)
WLAN_x_DIVERSITY_RX (Seite 254)
WLAN_x_DIVERSITY_TX (Seite 254)

Gelöschte Variablen

Abbildungsverzeichnis

3.1. Struktur des Paketfilters 42
3.2. Konfigurationsvariablen und ihre Anwendung 43
3.3. Der Weg von Paketen aus einem MASQ_NETWORK durch den Paketfilter . . 44
3.4. Der Weg von Paketen aus einem der TRUSTED_NETS bzw. ROUTE_NETWORK durch den Paketfilter 45
3.5. Der Weg von Paketen aus einem unbekannten Netz durch den Paketfilter . . . 47
3.6. Verzeichnisstruktur fli4l 55

4.1. VPN-Konfigurationsbeispiel — Tunnel zwischen zwei Routern 186
4.2. fli4l config Directory mit OpenVPN *.secret Dateien 190
4.3. Verbindungsübersicht 203
4.4. Detailansicht einer Verbindung (Keymanagement) 204
4.5. Beispiel 1 227
4.6. Beispiel 2 229
4.7. Beispiel 3 232
4.8. Verzeichnisstruktur fli4l 235

5.1. Einstellungen 267
5.2. Einstellungen für Diskette 268
5.3. Einstellungen für Remoteupdate 269
5.4. Einstellungen für HD-pre-install 270

8.1. Optionen für Dateien 304

Tabellenverzeichnis

3.1. Übersicht über die Zusatzpakete . . . 19
3.2. Automatische Einstellung der maximalen Verbindungsanzahl . . . 28
3.3. Tabelle der möglichen Netzwerkkartentreiber . . . 37
3.4. Tabelle der möglichen WLAN-Kartentreiber . . . 38
3.5. Ports, auf denen fli4l ggfs. Dienste anbietet . . . 47
3.6. Die im Lieferumfang von fli4l enthaltenen Templates . . . 57
3.7. Format der Imond-Logdatei . . . 79

4.1. Werte für BRIDGE_DEV_x_DEV_x_PATHCOST in Abhängigkeit von der Bandbreite . . . 95
4.2. Arten der pppoe-Paketerzeugung . . . 115
4.3. Bandbreite und CPU-Auslastung bei pppoe . . . 115
4.4. Fritz-Karten . . . 116
4.5. PPTP-Modemtypen . . . 117
4.6. Beispiel für die Konfiguration der Setup-Diskette . . . 127
4.7. Beispiel für eine Installation nach Typ A oder B . . . 127
4.9. Übersicht über mögliche Werte für LCD_TYPE_x . . . 165
4.10. Aktionen der OpenVPN-Webgui . . . 204
4.11. Unterschiedliche MTU Parameter der unterschiedlichen OpenVPN Versionen. . . . 206
4.12. Unterschiedliche MTU Parameter der fli4l-Router Versionen. . . . 206
4.13. OpenVPN Konfiguration mit 2 fli4l-Routern . . . 207
4.14. OpenVPN Konfiguration mit 2 fli4l-Routern deren Netzwerk über eine Funkverbindung gebridgt wird. . . . 207
4.15. OpenVPN Konfiguration mit 2 fli4l-Routern deren Netzwerk über eine Funkverbindung gebridgt wird. Die Konfiguration der Bridge in advanced_networking. 208
4.16. OpenVPN Konfiguration mit 2 fli4l-Routern deren Netzwerk über eine Funkverbindung gebridgt wird. Die Konfiguration der Bridge in der Basiskonfiguration (base.txt). . . . 208
4.17. OpenVPN Konfiguration mit einem Windowsrechner über GPRS. . . . 209
4.18. OpenVPN Absicherung eines WLAN. . . . 210

8.1. Parameter für mkfli4l . . . 302
8.2. Logische Ausdrücke . . . 320
8.3. Funktionen des cgi-helper Skriptes. . . . 333

Index

base.txt, 19
BEEP, 30
Beispiel-Datei(base.txt), 19
BONDING_DEV_N, 86
BONDING_DEV_x_ARP_INTERVAL, 90
BONDING_DEV_x_ARP_IP_TARGET_x, 90
BONDING_DEV_x_ARP_IP_TARGETS_N, 90
BONDING_DEV_x_DEV_N, 88
BONDING_DEV_x_DEV_x, 88
BONDING_DEV_x_DEVNAME, 86
BONDING_DEV_x_DOWNDELAY, 89
BONDING_DEV_x_LACP_RATE, 89
BONDING_DEV_x_MAC, 88
BONDING_DEV_x_MIIMON, 89
BONDING_DEV_x_MODE, 87
BONDING_DEV_x_PRIMARY, 89
BONDING_DEV_x_UPDELAY, 89
BONDING_DEV_x_USE_CARRIER, 89
BOOT_TYPE, 25
BOOTMENU_TIME, 27
BRIDGE_DEV_BOOTDELAY, 92
BRIDGE_DEV_N, 93
BRIDGE_DEV_x_AGING, 93
BRIDGE_DEV_x_DEV_N, 93
BRIDGE_DEV_x_DEV_x_DEV, 93
BRIDGE_DEV_x_DEV_x_PATHCOST, 95
BRIDGE_DEV_x_DEV_x_PORT_PRIORITY, 95
BRIDGE_DEV_x_DEVNAME, 93
BRIDGE_DEV_x_FORWARD_DELAY, 94
BRIDGE_DEV_x_GARBAGE_COLLECTION_INTERVAL, 93
BRIDGE_DEV_x_HELLO, 94
BRIDGE_DEV_x_MAX_MESSAGE_AGE, 95
BRIDGE_DEV_x_NAME, 93
BRIDGE_DEV_x_PRIORITY, 94
BRIDGE_DEV_x_STP, 93
BUILDDIR, 271

CHRONY_BIOS_TIME, 99
CHRONY_LOG, 99
CHRONY_TIMESERVER_N, 99
CHRONY_TIMESERVER_x, 99
CHRONY_TIMESERVICE, 99
COMP_TYPE_KERNEL, 28
COMP_TYPE_OPT, 28
COMP_TYPE_ROOTFS, 28
CONSOLE_BLANK_TIME, 30

DEBUG_ENABLE_CORE, 328
DEBUG_IP, 328
DEBUG_IPUP, 328
DEBUG_KEEP_BOOTLOGD, 328
DEBUG_MDEV, 328
DEBUG_MODULES, 31
DEBUG_STARTUP, 31
DENY_ICMP, 48
DEV_MTU_N, 92
DEV_MTU_x, 92
DHCP_CLIENT_DEBUG, 101
DHCP_CLIENT_DEV, 379
DHCP_CLIENT_HOSTNAME, 379
DHCP_CLIENT_IGNORE_DEF_GW, 379
DHCP_CLIENT_INTERFACES, 379
DHCP_CLIENT_N, 100
DHCP_CLIENT_TYPE, 100
DHCP_CLIENT_USEPEERDNS, 379
DHCP_CLIENT_x_HOSTNAME, 101
DHCP_CLIENT_x_IF, 100
DHCP_CLIENT_x_ROUTE, 101

DHCP_CLIENT_x_USEPEERDNS, 101
DHCP_DENY_MAC_N, 108
DHCP_DENY_MAC_x, 108
DHCP_EXTRA_RANGE_N, 107
DHCP_EXTRA_RANGE_x_DEVICE, 107
DHCP_EXTRA_RANGE_x_DNS_SERVER, 107
DHCP_EXTRA_RANGE_x_END, 107
DHCP_EXTRA_RANGE_x_GATEWAY, 107
DHCP_EXTRA_RANGE_x_MTU, 107
DHCP_EXTRA_RANGE_x_NETMASK, 107
DHCP_EXTRA_RANGE_x_NTP_SERVER, 107
DHCP_EXTRA_RANGE_x_START, 107
DHCP_LEASES_DIR, 106
DHCP_LEASES_VOLATILE, 106
DHCP_LS_TIME_DYN, 106
DHCP_LS_TIME_FIX, 106
DHCP_MAX_LS_TIME_DYN, 106
DHCP_MAX_LS_TIME_FIX, 106
DHCP_RANGE_N, 106
DHCP_RANGE_x_DNS_DOMAIN, 107
DHCP_RANGE_x_DNS_SERVER1, 106
DHCP_RANGE_x_DNS_SERVER2, 107
DHCP_RANGE_x_END, 106
DHCP_RANGE_x_GATEWAY, 107
DHCP_RANGE_x_MTU, 107
DHCP_RANGE_x_NET, 106
DHCP_RANGE_x_NTP_SERVER, 107
DHCP_RANGE_x_OPTION_N, 107
DHCP_RANGE_x_OPTION_x, 107
DHCP_RANGE_x_PXE_FILENAME, 108
DHCP_RANGE_x_PXE_OPTIONS, 108
DHCP_RANGE_x_PXE_SERVERIP, 108
DHCP_RANGE_x_PXE_SERVERNAME, 108
DHCP_RANGE_x_START, 106
DHCP_TYPE, 106
DHCP_VERBOSE, 106
DHCP_WINSSERVER_1, 106
DHCP_WINSSERVER_2, 106
DHCPRELAY_IF_N, 109
DHCPRELAY_IF_x, 109
DHCPRELAY_SERVER, 108
DIALMODE, 80
DMZ_LOG, 72
DMZ_NAT, 72
DMZ_ORANGE_RED_N, 72
DMZ_ORANGE_RED_x, 72
DMZ_ORANGE_ROUTER_N, 72
DMZ_ORANGE_ROUTER_x, 72
DMZ_RED_DEV, 71
DNS_BOGUS_PRIV, 104
DNS_FILTERWIN2K, 104
DNS_FORBIDDEN_N, 103
DNS_FORBIDDEN_x, 103
DNS_FORWARDERS, 77
DNS_LISTENIP_N, 103
DNS_LISTENIP_x, 103
DNS_MX_SERVER, 103
DNS_REBINDOK_N, 105
DNS_REBINDOK_x_DOMAIN, 105
DNS_REDIRECT_N, 104
DNS_REDIRECT_x, 104
DNS_REDIRECT_x_IP, 104
DNS_SPECIAL_N, 104
DNS_SPECIAL_x, 104
DNS_SPECIAL_x_DNSIP, 104
DNS_SPECIAL_x_DOMAIN, 104
DNS_SPECIAL_x_NETWORK, 104
DNS_SUPPORT_IPV6, 104
DNS_VERBOSE, 103
DOMAIN_NAME, 77
DRIVE, 271
DYNDNS_ALLOW_SSL, 123
DYNDNS_DEBUG_PROVIDER, 123
DYNDNS_LOOKUP_NAMES, 123
DYNDNS_N, 122
DYNDNS_NO_FRAMES, 379
DYNDNS_SAVE_OUTPUT, 122
DYNDNS_x_CIRCUIT, 122
DYNDNS_x_EXTIP, 123
DYNDNS_x_HOSTNAME, 122
DYNDNS_x_PASSWORD, 122
DYNDNS_x_PROVIDER, 122
DYNDNS_x_RENEW, 123
DYNDNS_x_USER, 122

EASYCRON_MAIL, 123
EASYCRON_N, 123
EASYCRON_x_COMMAND, 124
EASYCRON_x_CUSTOM, 124
EASYCRON_x_TIME, 124
EXTMOUNT_N, 128
EXTMOUNT_x_FILESYSTEM, 128
EXTMOUNT_x_MOUNTPOINT, 129
EXTMOUNT_x_OPTIONS, 129
EXTMOUNT_x_VOLUMEID, 128

FILESONLY, 271
FLOPPYMODE, 271
FORWARD_DENY_PORT_N, 46
FORWARD_DENY_PORT_x, 46
FORWARD_HOST_N, 46
FORWARD_HOST_WHITE, 45
FORWARD_HOST_x, 46
FRITZDSL_CHARGEINT, 111
FRITZDSL_DEBUG, 111
FRITZDSL_FILTER, 113
FRITZDSL_HUP_TIMEOUT, 111
FRITZDSL_MRU, 113
FRITZDSL_MTU, 113
FRITZDSL_NAME, 110
FRITZDSL_NF_MSS, 113
FRITZDSL_PASS, 111
FRITZDSL_PROVIDER, 116
FRITZDSL_TIMES, 111
FRITZDSL_TYPE, 116
FRITZDSL_USEPEERDNS, 110
FRITZDSL_USER, 111
ftp, 75

HDDRV_N, 130
HDDRV_x, 130
HDDRV_x_OPTION, 130
HDSLEEP_TIMEOUT, 129
HOST_EXTRA_x_IP4, 102
HOST_EXTRA_x_IP6, 102
HOST_EXTRA_x_NAME, 102
HOST_x_ALIAS_N, 101
HOST_x_ALIAS_x, 101
HOST_x_DHCPTYP, 101
HOST_x_DOMAIN, 101
HOST_x_IP4, 101
HOST_x_IP6, 101
HOST_x_MAC, 101
HOST_x_NAME, 101
HOST_x_PXE_FILENAME, 108
HOST_x_PXE_OPTIONS, 108
HOST_x_PXE_SERVERIP, 108
HOST_x_PXE_SERVERNAME, 108
HOSTNAME, 25
HOSTNAME_ALIAS_N, 77
HOSTNAME_ALIAS_x, 77
HOSTNAME_IP, 77
HOSTNAME_IP6, 137
HOSTS_EXTRA_N, 102
HOSTS_GLOBAL, 272
HOSTS_N, 101
HTTPD_ARPING, 132
HTTPD_GUI_LANG, 132
HTTPD_GUI_SKIN, 132
HTTPD_LISTENIP, 132
HTTPD_PORT, 132
HTTPD_PORTFW, 132
HTTPD_USER, 365
HTTPD_USER_N, 133
HTTPD_USER_x_PASSWORD, 133
HTTPD_USER_x_RIGHTS, 133
HTTPD_USER_x_USERNAME, 133
HW_DETECT_AT_BOOTTIME, 240

IGNORE_KSOFTIRQ, 379
IMOND_ADMIN_PASS, 78
IMOND_BEEP, 79
IMOND_DIAL, 80
IMOND_ENABLE, 80
IMOND_LED, 78
IMOND_LOG, 79
IMOND_LOGDIR, 79
IMOND_PASS, 78
IMOND_PORT, 78
IMOND_REBOOT, 80
IMOND_ROUTE, 80
IMOND_USE_ORIG, 78
INPUT_ACCEPT_PORT_N, 48
INPUT_ACCEPT_PORT_x, 48
INPUT_POLICY, 48
IP_CONNTRACK_MAX, 28
IP_DYN_ADDR, 80
IP_NET_N, 38
IP_NET_x, 38

IP_NET_x_COMMENT, 40
IP_NET_x_DEV, 39
IP_NET_x_MAC, 40
IP_NET_x_NAME, 40
IP_NET_x_TYPE, 40, 71
IP_ROUTE_N, 40
IP_ROUTE_x, 40
IPV6_MULTICAST, 141
IPV6_NET_N, 137
IPV6_NET_x, 137
IPV6_NET_x_ADVERTISE, 137
IPV6_NET_x_ADVERTISE_DNS, 138
IPV6_NET_x_DEV, 137
IPV6_NET_x_DHCP, 138
IPV6_NET_x_NAME, 138
IPV6_ROUTE_N, 140
IPV6_ROUTE_x, 140
IPV6_TUNNEL_N, 139
IPV6_TUNNEL_x_DEV, 139
IPV6_TUNNEL_x_LOCALV4, 139
IPV6_TUNNEL_x_LOCALV6, 139
IPV6_TUNNEL_x_MTU, 139
IPV6_TUNNEL_x_REMOTEV4, 139
IPV6_TUNNEL_x_SIXXS_ID, 140
IPV6_TUNNEL_x_SIXXS_-
PASSWORD, 140
IPV6_TUNNEL_x_SIXXS_-
USERNAME, 140
IPV6_TUNNEL_x_TYPE, 140
irc, 75
ISDN_CIRC_x_AUTH, 157
ISDN_CIRC_x_BANDWIDTH, 152
ISDN_CIRC_x_BUNDLING, 152
ISDN_CIRC_x_CALLBACK, 156
ISDN_CIRC_x_CBDELAY, 156
ISDN_CIRC_x_CBNUMBER, 156
ISDN_CIRC_x_CHARGEINT, 158
ISDN_CIRC_x_CLAMP_MSS, 154
ISDN_CIRC_x_DEBUG, 157
ISDN_CIRC_x_DIALIN, 155
ISDN_CIRC_x_DIALOUT, 155
ISDN_CIRC_x_EAZ, 157
ISDN_CIRC_x_FRAMECOMP, 154
ISDN_CIRC_x_HEADERCOMP, 154
ISDN_CIRC_x_HUP_TIMEOUT, 157
ISDN_CIRC_x_LOCAL, 153
ISDN_CIRC_x_MRU, 153
ISDN_CIRC_x_MTU, 153
ISDN_CIRC_x_NAME, 151
ISDN_CIRC_x_PASS, 155
ISDN_CIRC_x_REMOTE, 153
ISDN_CIRC_x_REMOTENAME, 154
ISDN_CIRC_x_ROUTE_N, 155
ISDN_CIRC_x_ROUTE_X, 155
ISDN_CIRC_x_SLAVE_EAZ, 157
ISDN_CIRC_x_TIMES, 158
ISDN_CIRC_x_TYPE, 152
ISDN_CIRC_x_USEPEERDNS, 151
ISDN_CIRC_x_USER, 155
ISDN_CIRCUITS_N, 151
ISDN_DEBUG_LEVEL, 149
ISDN_FILTER, 150
ISDN_IO, 146
ISDN_IO0, 146
ISDN_IO1, 146
ISDN_LZS_COMP, 150
ISDN_LZS_DEBUG, 150
ISDN_LZS_TWEAK, 150
ISDN_MEM, 146
ISDN_TYPE, 146
ISDN_VERBOSE_LEVEL, 149

KERNEL_BOOT_OPTION, 28
KERNEL_VERSION, 27
KEYBOARD_LOCALE, 32

LCD_ADDR_TYPE, 163
LCD_ADDRESS, 162
LCD_COLS, 162
LCD_DSL_SPEED_IN, 166
LCD_DSL_SPEED_OUT, 166
LCD_FILTER, 164
LCD_LANIP, 163
LCD_LANPASS, 163
LCD_LANTYPE, 163
LCD_LANUSER, 163
LCD_LINES, 162
LCD_REBOOT_MSG, 164
LCD_START_ISDN_RATE, 164
LCD_START_MSG, 164
LCD_STOP_MSG, 164
LCD_TIME_LONG, 163
LCD_TIME_SHORT, 163
LCD_TYPE_N, 164

LCD_TYPE_OFFLINE_N, 166
LCD_TYPE_OFFLINE_x, 166
LCD_TYPE_ONLINE_N, 166
LCD_TYPE_ONLINE_x, 166
LCD_TYPE_x, 164
LCD_WINAMP, 164
LOCALE, 29
LOG_BOOT_SEQ, 328
LOGIP_LOGDIR, 83
LPDSRV_NETWORK_N, 173
LPDSRV_NETWORK_x, 173
LPDSRV_PARPORT_N, 174
LPDSRV_PARPORT_x_DMA, 175
LPDSRV_PARPORT_x_IO, 174
LPDSRV_PARPORT_x_IRQ, 175
LPDSRV_REMOTE_N, 177
LPDSRV_REMOTE_x_IP, 178
LPDSRV_REMOTE_x_PORTNR, 178
LPDSRV_USBPORT_N, 176

MASQ_MODULE_N, 75
MASQ_MODULE_x, 75
MASQ_MODULE_x_OPTION, 75
MASQ_NETWORK, 42
Masquerading, 75
mIRC, 76
MKFLI4L_DEBUG_OPTION, 272
MOUNT_BOOT, 26

NET_DRV_N, 32
NET_DRV_x, 32
NET_DRV_x_OPTION, 32

OAC_ALL_INVISIBLE, 134
OAC_BLOCK_UNKNOWN_IF_x, 135
OAC_GROUP_N, 135
OAC_GROUP_x_BOOTBLOCK, 135
OAC_GROUP_x_CLIENT_N, 135
OAC_GROUP_x_CLIENT_x, 135
OAC_GROUP_x_INVISIBLE, 135
OAC_GROUP_x_NAME, 135
OAC_INPUT, 134
OAC_LIMITS, 134
OAC_MODE, 135
OAC_WANDEVICE, 134
OPENVPN_DEFAULT_ALLOW_-ICMPPING, 194
OPENVPN_DEFAULT_CIPHER, 193
OPENVPN_DEFAULT_COMPRESS, 193
OPENVPN_DEFAULT_CREATE_-SECRET, 194
OPENVPN_DEFAULT_DIGEST, 194
OPENVPN_DEFAULT_FLOAT, 194
OPENVPN_DEFAULT_FRAGMENT, 197
OPENVPN_DEFAULT_KEYSIZE, 194
OPENVPN_DEFAULT_LINK_MTU, 197
OPENVPN_DEFAULT_-MANAGEMENT_LOG_-CACHE, 197
OPENVPN_DEFAULT_-MANAGMENT_LOG_CACHE, 381
OPENVPN_DEFAULT_MSSFIX, 197
OPENVPN_DEFAULT_MUTE_-REPLAY_WARNINGS, 197
OPENVPN_DEFAULT_OPEN_-OVPNPORT, 194
OPENVPN_DEFAULT_PF_DMZ_-TYPE, 195
OPENVPN_DEFAULT_PF_-FORWARD_LOG, 195
OPENVPN_DEFAULT_PF_-FORWARD_POLICY, 195
OPENVPN_DEFAULT_PF_INPUT_-LOG, 195
OPENVPN_DEFAULT_PF_INPUT_-POLICY, 195
OPENVPN_DEFAULT_PING, 195
OPENVPN_DEFAULT_PING_-RESTART, 195
OPENVPN_DEFAULT_PROTOCOL, 196
OPENVPN_DEFAULT_RESOLV_-RETRY, 195
OPENVPN_DEFAULT_RESTART, 196
OPENVPN_DEFAULT_SHAPER, 197
OPENVPN_DEFAULT_START, 196
OPENVPN_DEFAULT_TUN_MTU, 197
OPENVPN_DEFAULT_TUN_MTU_-EXTRA, 197
OPENVPN_DEFAULT_VERBOSE, 196
OPENVPN_EXPERT, 198

OPENVPN_FEATURES, 381
OPENVPN_N, 198
OPENVPN_WEBGUI, 202
OPENVPN_x_ACTIV, 198
OPENVPN_x_ALLOW_ICMPPING, 200
OPENVPN_x_BRIDGE, 190
OPENVPN_x_BRIDGE_COST, 190
OPENVPN_x_BRIDGE_PRIORITY, 191
OPENVPN_x_CHECK_CONFIG, 198
OPENVPN_x_CIPHER, 199
OPENVPN_x_COMPRESS, 199
OPENVPN_x_CREATE_SECRET, 199
OPENVPN_x_DIGEST, 199
OPENVPN_x_DNSIP, 193
OPENVPN_x_DNSIP_x, 193
OPENVPN_x_DOMAIN, 193
OPENVPN_x_DOMAIN_x, 193
OPENVPN_x_FLOAT, 199
OPENVPN_x_FRAGMENT, 202
OPENVPN_x_ISDN_CIRC_NAME, 199
OPENVPN_x_KEYSIZE, 199
OPENVPN_x_LINK_MTU, 202
OPENVPN_x_LOCAL_HOST, 188
OPENVPN_x_LOCAL_PORT, 188
OPENVPN_x_LOCAL_VPN_IP, 191
OPENVPN_x_MANAGEMENT_LOG_-CACHE, 200
OPENVPN_x_MANAGMENT_LOG_-CACHE, 381
OPENVPN_x_MANAGMENT_PORT, 381
OPENVPN_x_MSSFIX, 202
OPENVPN_x_MUTE_REPLAY_-WARNINGS, 200
OPENVPN_x_NAME, 198
OPENVPN_x_OPEN_OVPNPORT, 200
OPENVPN_x_PF_DMZ_TYPE, 200
OPENVPN_x_PF_FORWARD_LOG, 201
OPENVPN_x_PF_FORWARD_N, 201
OPENVPN_x_PF_FORWARD_-POLICY, 201
OPENVPN_x_PF_FORWARD_x, 201
OPENVPN_x_PF_INPUT_LOG, 200
OPENVPN_x_PF_INPUT_N, 201
OPENVPN_x_PF_INPUT_POLICY, 200
OPENVPN_x_PF_INPUT_x, 201
OPENVPN_x_PF_POSTROUTING_N, 201
OPENVPN_x_PF_POSTROUTING_x, 201
OPENVPN_x_PF_PREROUTING_N, 201
OPENVPN_x_PF_PREROUTING_x, 201
OPENVPN_x_PING, 199
OPENVPN_x_PING_RESTART, 200
OPENVPN_x_PROTOCOL, 199
OPENVPN_x_REMOTE_HOST, 187
OPENVPN_x_REMOTE_HOST_x, 188
OPENVPN_x_REMOTE_HOSTS_N, 187
OPENVPN_x_REMOTE_PORT, 188
OPENVPN_x_REMOTE_VPN_IP, 191
OPENVPN_x_RESOLV_RETRY, 199
OPENVPN_x_RESTART, 200
OPENVPN_x_ROUTE_N, 192
OPENVPN_x_ROUTE_x, 192
OPENVPN_x_SECRET, 189
OPENVPN_x_SHAPER, 202
OPENVPN_x_START, 200
OPENVPN_x_TUN_MTU, 202
OPENVPN_x_TUN_MTU_EXTRA, 202
OPENVPN_x_TYPE, 189
OPENVPN_x_VERBOSE, 200
OPT_ARP, 240
OPT_BCRELAY, 240
OPT_BONDING_DEV, 86
OPT_BRIDGE_DEV, 92
OPT_CHRONY, 99
OPT_DHCP, 106
OPT_DHCP_CLIENT, 100
OPT_DHCPRELAY, 108
OPT_DMZ, 71
OPT_DNS, 103
OPT_DYNDNS, 122
OPT_E3, 240
OPT_EASYCRON, 123
OPT_EBTABLES, 96
OPT_EXTMOUNT, 128
OPT_FRITZDSL, 116

OPT_FTP, 241
OPT_HDDRV, 130
OPT_HDINSTALL, 125
OPT_HDSLEEP, 129
OPT_HTTPD, 131
OPT_HW_DETECT, 240
OPT_IFTOP, 241
OPT_IMONC, 241
OPT_IPERF, 241
OPT_IPV6, 137
OPT_ISDN, 145
OPT_ISDN_COMP, 150
OPT_KLOGD, 83
OPT_LCD, 162
OPT_LESS, 381
OPT_LOGIP, 83
OPT_LPDSRV, 172
OPT_LPDSRV_PARPORT, 173
OPT_LPDSRV_REMOTE, 177
OPT_LPDSRV_USBPORT, 176
OPT_LSPCI, 241
OPT_MAKEKBL, 32
OPT_MOUNT, 128
OPT_MOUNTFLOPPY, 29
OPT_MTOOLS, 242
OPT_NETCAT, 242
OPT_NETSTAT, 242
OPT_NTTCP, 242
OPT_OAC, 134
OPT_OPENVPN, 187
OPT_PCMCIA, 210
OPT_PFC, 118
OPT_PLINK_CLIENT, 238
OPT_PNP, 84
OPT_POESTATUS, 118
OPT_PPP, 211
OPT_PPPOE, 114
OPT_PPPOE_CIRC, 115
OPT_PPTP, 117
OPT_PRIVOXY, 213
OPT_QOS, 219
OPT_RECOVER, 129
OPT_RTMON, 243
OPT_SCP, 238
OPT_SERIAL, 243
OPT_SFTPSERVER, 239
OPT_SHRED, 243
OPT_SS5, 217
OPT_SSH_CLIENT, 238
OPT_SSHD, 234
OPT_STRACE, 243
OPT_SYSLOGD, 81
OPT_TAIL, 381
OPT_TCPDUMP, 243
OPT_TELMOND, 159
OPT_TFTP, 109
OPT_TOP, 381
OPT_TOR, 216
OPT_TRACEROUTE, 243
OPT_TRACEROUTE6, 244
OPT_TRANSPROXY, 218
OPT_UMTS, 244
OPT_USB, 247
OPT_VALGRIND, 244
OPT_VLAN_DEV, 91
OPT_WGET, 244
OPT_WLAN, 250
OPT_Y2K, 83
OPT_YTREE, 244

PACKETFILTER_LOG, 48
PACKETFILTER_LOG_LEVEL, 49
PASSWORD, 25
PCMCIA_CARDMGR_OPTS, 211
PCMCIA_CORE_OPTS, 211
PCMCIA_MISC_N, 211
PCMCIA_MISC_x, 211
PCMCIA_PCIC, 210
PCMCIA_PCIC_EXTERN, 211
PCMCIA_PCIC_OPTS, 210
PF6_FORWARD_ACCEPT_DEF, 143
PF6_FORWARD_LOG, 143
PF6_FORWARD_LOG_LIMIT, 143
PF6_FORWARD_N, 143
PF6_FORWARD_POLICY, 143
PF6_FORWARD_REJ_LIMIT, 143
PF6_FORWARD_UDP_REJ_LIMIT, 143
PF6_FORWARD_x, 144
PF6_FORWARD_x_COMMENT, 144
PF6_INPUT_ACCEPT_DEF, 141
PF6_INPUT_LOG, 141
PF6_INPUT_LOG_LIMIT, 142
PF6_INPUT_N, 142

PF6_INPUT_POLICY, 141
PF6_INPUT_REJ_LIMIT, 142
PF6_INPUT_UDP_REJ_LIMIT, 142
PF6_INPUT_x, 142
PF6_INPUT_x_COMMENT, 142
PF6_USR_CHAIN_N, 144
PF6_USR_CHAIN_x_NAME, 144
PF6_USR_CHAIN_x_RULE_N, 144
PF6_USR_CHAIN_x_RULE_x, 144
PF6_USR_CHAIN_x_RULE_x_-
COMMENT, 144
PF_FORWARD_ACCEPT_DEF, 59
PF_FORWARD_LOG, 60
PF_FORWARD_LOG_LIMIT, 60
PF_FORWARD_N, 60
PF_FORWARD_POLICY, 59
PF_FORWARD_REJ_LIMIT, 60
PF_FORWARD_UDP_REJ_LIMIT, 60
PF_FORWARD_x, 60
PF_FORWARD_x_COMMENT, 60
PF_INPUT_ACCEPT_DEF, 57
PF_INPUT_ICMP_ECHO_REQ_-
LIMIT, 59
PF_INPUT_LOG, 58
PF_INPUT_LOG_LIMIT, 58
PF_INPUT_N, 58
PF_INPUT_POLICY, 57
PF_INPUT_REJ_LIMIT, 58
PF_INPUT_UDP_REJ_LIMIT, 58
PF_INPUT_x, 58
PF_INPUT_x_COMMENT, 58
PF_LOG_LEVEL, 57
PF_NEW_CONFIG, 41
PF_POSTROUTING_N, 61
PF_POSTROUTING_x, 61
PF_POSTROUTING_x_COMMENT, 61
PF_PREROUTING_N, 62
PF_PREROUTING_x, 62
PF_PREROUTING_x_COMMENT, 62
PF_USR_CHAIN_N, 60
PF_USR_CHAIN_x_NAME, 60
PF_USR_CHAIN_x_RULE_N, 60
PF_USR_CHAIN_x_RULE_x, 60
PF_USR_CHAIN_x_RULE_x_-
COMMENT, 60
PORTFW_N, 73
PORTFW_x_COMMENT, 74
PORTFW_x_NEW_TARGET, 74
PORTFW_x_PROTOCOL, 74
PORTFW_x_TARGET, 73
POWERMANAGEMENT, 28
POWEROFF_ON_HALT, 379
PPP_DEV, 211
PPP_IPADDR, 211
PPP_NETMASK, 211
PPP_NETWORK, 211
PPP_PEER, 211
PPP_SPEED, 211
PPPOE_CHARGEINT, 111
PPPOE_CIRC_N, 115
PPPOE_CIRC_x_CHARGEINT, 115
PPPOE_CIRC_x_DEBUG, 115
PPPOE_CIRC_x_ETH, 115
PPPOE_CIRC_x_FILTER, 115
PPPOE_CIRC_x_HUP_TIMEOUT, 115
PPPOE_CIRC_x_MRU, 115
PPPOE_CIRC_x_MTU, 115
PPPOE_CIRC_x_NAME, 115
PPPOE_CIRC_x_PASS, 115
PPPOE_CIRC_x_TIMES, 115
PPPOE_CIRC_x_TYPE, 115
PPPOE_CIRC_x_USEPEERDNS, 115
PPPOE_CIRC_x_USER, 115
PPPOE_DEBUG, 111
PPPOE_ETH, 114
PPPOE_FILTER, 113
PPPOE_HUP_TIMEOUT, 111, 115
PPPOE_MRU, 113
PPPOE_MTU, 113
PPPOE_NAME, 110
PPPOE_NF_MSS, 113
PPPOE_PASS, 111
PPPOE_TIMES, 111
PPPOE_TYPE, 114
PPPOE_USEPEERDNS, 110
PPPOE_USER, 111
pptp, 76
PPTP_CHARGEINT, 111
PPTP_CLIENT_LOGLEVEL, 118
PPTP_CLIENT_REORDER_TO, 117
PPTP_DEBUG, 111
PPTP_ETH, 117
PPTP_FILTER, 113
PPTP_HUP_TIMEOUT, 111

PPTP_MODEM_TYPE, 117
PPTP_NAME, 110
PPTP_PASS, 111
PPTP_TIMES, 111
PPTP_USEPEERDNS, 110
PPTP_USER, 111
PRESERVE, 26
PRIVOXY_MENU, 214
PRIVOXY_N, 214
PRIVOXY_x_ACTIONDIR, 214
PRIVOXY_x_ALLOW_N, 214
PRIVOXY_x_ALLOW_x, 214
PRIVOXY_x_CONFIG, 215
PRIVOXY_x_HTTP_PROXY, 215
PRIVOXY_x_LISTEN, 214
PRIVOXY_x_LOGDIR, 215
PRIVOXY_x_LOGLEVEL, 215
PRIVOXY_x_SOCKS_PROXY, 215
PRIVOXY_x_TOGGLE, 215
PXESUBDIR, 272

QOS_CLASS_N, 220
QOS_CLASS_x_DIRECTION, 222
QOS_CLASS_x_MAXBANDWIDTH, 222
QOS_CLASS_x_MINBANDWIDTH, 221
QOS_CLASS_x_PARENT, 221
QOS_CLASS_x_PRIO, 222
QOS_FILTER_N, 223
QOS_FILTER_x_CLASS, 223
QOS_FILTER_x_IP_EXTERN, 224
QOS_FILTER_x_IP_INTERN, 224
QOS_FILTER_x_OPTION, 225
QOS_FILTER_x_PORT, 224
QOS_FILTER_x_PORT_TYPE, 225
QOS_INTERNET_BAND_DOWN, 219
QOS_INTERNET_BAND_UP, 220
QOS_INTERNET_DEFAULT_DOWN, 220
QOS_INTERNET_DEFAULT_UP, 220
QOS_INTERNET_DEV_N, 219
QOS_INTERNET_DEV_x, 219

REAL_MODE_POWEROFF, 379
REMOTEHOSTNAME, 271
REMOTEPATHNAME, 272
REMOTEPORT, 272
REMOTEUPDATE, 271
REMOTEUSERNAME, 271
ROUTE_NETWORK, 44

SER_BEEP, 379
SER_CONSOLE, 30
SER_CONSOLE_IF, 30
SER_CONSOLE_RATE, 30
SQUEEZE_SCRIPTS, 272
SS5_ALLOW_N, 218
SS5_ALLOW_x, 218
SS5_LISTEN_N, 217
SS5_LISTEN_x, 217
SSHD_ALLOWPASSWORDLOGIN, 234
SSHD_CREATEHOSTKEYS, 234
SSHD_PORT, 236
SSHD_PRIVATE_KEYFILE_x, 237
SSHD_PRIVATE_KEYFILES_N, 237
SSHD_PUBLIC_KEY_x, 237
SSHD_PUBLIC_KEYFILE_x, 237
SSHD_PUBLIC_KEYFILES_N, 237
SSHD_PUBLIC_KEYS_N, 236
SSHKEYFILE, 272
START_IMOND, 78
SYSLOGD_DEST_N, 81
SYSLOGD_DEST_x, 81
SYSLOGD_RECEIVER, 81
SYSLOGD_ROTATE, 83
SYSLOGD_ROTATE_DIR, 83

TELMOND_CMD_N, 161
TELMOND_CMD_x, 161
TELMOND_LOG, 160
TELMOND_LOGDIR, 160
TELMOND_MSN_N, 160
TELMOND_MSN_x, 160
TELMOND_PORT, 160
TFTP_PATH, 109
TFTPBOOTIMAGE, 272
TFTPBOOTPATH, 272
TIME_INFO, 27
TOR_ALLOW_N, 217
TOR_ALLOW_x, 217
TOR_CONTROL_PASSWORD, 217
TOR_CONTROL_PORT, 217
TOR_DATA_DIR, 217
TOR_HTTP_PROXY, 217

TOR_HTTP_PROXY_AUTH, 217
TOR_HTTPS_PROXY, 217
TOR_HTTPS_PROXY_AUTH, 217
TOR_LISTEN_N, 216
TOR_LISTEN_x, 216
TOR_LOGFILE, 217
TOR_LOGLEVEL, 217
TRANSPROXY_ALLOW_N, 218
TRANSPROXY_ALLOW_x, 219
TRANSPROXY_LISTEN_N, 218
TRANSPROXY_LISTEN_x, 218
TRANSPROXY_TARGET_IP, 218
TRANSPROXY_TARGET_PORT, 218
TRUSTED_NETS, 46

UMTS_ADAPTER, 246
UMTS_APN, 245
UMTS_CHARGEINT, 245
UMTS_CTRL, 247
UMTS_DEBUG, 244
UMTS_DEV, 246
UMTS_DIALOUT, 244
UMTS_DRV, 246
UMTS_FILTER, 245
UMTS_GPRS_UMTS, 244
UMTS_HUP_TIMEOUT, 245
UMTS_IDDEVICE, 246
UMTS_IDDEVICE2, 246
UMTS_IDVENDOR, 246
UMTS_IDVENDOR2, 246
UMTS_NAME, 245
UMTS_PASSWD, 245
UMTS_PIN, 244
UMTS_SWITCH, 246
UMTS_TIMES, 245
UMTS_USEPEERDNS, 245
UMTS_USER, 245
USB_EXTRA_DRIVER_N, 248
USB_EXTRA_DRIVER_x, 248
USB_EXTRA_DRIVER_x_PARAM, 248
USB_LOWLEVEL, 247
USB_MODEM_WAITSECONDS, 248
USBMODEM_WAITSECONDS, 378, 382

VERBOSE, 271
VLAN_DEV_N, 91
VLAN_DEV_x_DEV, 91
VLAN_DEV_x_VID, 91

WGET_SSL, 244
WLAN_N, 250
WLAN_REGDOMAIN, 250
WLAN_WEBGUI, 250
WLAN_x_ACL_MAC_N, 254
WLAN_x_ACL_MAC_x, 254
WLAN_x_ACL_POLICY, 254
WLAN_x_AP, 254
WLAN_x_CHANNEL, 251
WLAN_x_DIVERSITY, 254
WLAN_x_DIVERSITY_RX, 254
WLAN_x_DIVERSITY_TX, 254
WLAN_x_ENC_ACTIVE, 252
WLAN_x_ENC_MODE, 252
WLAN_x_ENC_N, 252
WLAN_x_ENC_x, 252
WLAN_x_ESSID, 251
WLAN_x_MAC, 250
WLAN_x_MACOVERRIDE, 250
WLAN_x_MODE, 251
WLAN_x_NOESSID, 251
WLAN_x_RATE, 251
WLAN_x_RTS, 252
WLAN_x_WEP_ROTATE, 253
WLAN_x_WPA_DEBUG, 253
WLAN_x_WPA_ENCRYPTION, 253
WLAN_x_WPA_KEY_MGMT, 253
WLAN_x_WPA_PSK, 253
WLAN_x_WPA_TYPE, 253

Y2K_DAYS, 84